SOLUTIONS CUBED

MICRO PID Position Controller Data Sheet

Revision 2
February 10th, 2003

Copyright © 2002 Solutions Cubed

Table of Contents

1. Revision Log **3**

2. Introduction **4**
2.1 Description 4

3. Engineering Specifications **5**
3.1 Absolute Maximum Ratings 5
3.2 DC Electrical Characteristics 5
3.3 AC Electrical Characteristics 6
3.4 Mechanical Dimensions 7
3.5 Connectivity Overview 8
3.6 Power Supply Pin Connections 9
3.7 PID Filter Configuration and Settings 10
3.8 PID Filter Tuning 14
3.9 Trapezoidal Movement Profiles 14
3.10 Step Logger Mode and Trap Logger Mode 17
3.11 4X_1X Mode 18
3.12 ANALOG_IN, CHA, and CHB Input Circuits 18
3.13 Two's Compliment Number System 18
3.14 RS232 Conversion Circuitry 19
3.15 Limit Switch Functionality 20
3.16 VM Capacitor 20
3.17 Fault Conditions 20

4. Operating Information **21**
4.1 Overview 21
4.2 Analog Control Mode 21
4.3 Trapezoidal Control Mode 22
4.4 Velocity Control Mode 24
4.5 Direct Control Mode 26

5. Communication Protocol **27**
5.1 Overview 27
5.2 Command Set 27
5.3 USER0, USER1, and STATUS Flags 36
5.4 Changing Baud Rates 38

6. Quick Start **39**

List of Figures

Figure 1: Mechanical Dimensions	7
Figure 2: Mechanical Landmark Descriptions	7
Figure 3: MICRO PID Position Controller Pin Definitions	8
Figure 4: Power Supply Pin Connections	9
Figure 5: PID Programming Window for Solutions Cubed Software	10
Figure 6: Step Function – “P” Only	11
Figure 7: Step Function – “P” And “I” Only	12
Figure 8: Step Function – “P”, “I”, And “D”	13
Figure 9: PID Period Constants	13
Figure 10: Trapezoidal Profile Programming Window for Solutions Cubed Software	15
Figure 11: Components of a Trapezoidal Segment	15
Figure 12: Velocity Graph of Trapezoidal Profile	16
Figure 13: Trapezoidal Landing Adjustments	16
Figure 14: Position Graph of Trapezoidal Profile	17
Figure 15: ANALOG_IN, CHA, and CHB Input Circuits	18
Figure 16: Two’s Compliment Examples	18
Figure 17: RS232 <-> TTL Conversion	19
Figure 18: Protection for Limit Switch Inputs	19
Figure 19: Registers and Bits Associated with the Limit Switches	20
Figure 20: MICRO PID Position Controller Analog Mode Connections	22
Figure 21: Movement Segment Components	23
Figure 22: Movement Profile Components	23
Figure 23: “Write Velocity Settings” Components	25
Figure 24: Position Controller Command Set	27
Figure 25: USER0 Bits	36
Figure 26: USER1 Bits	37
Figure 27: STATUS Bits	37

1. Revision Log

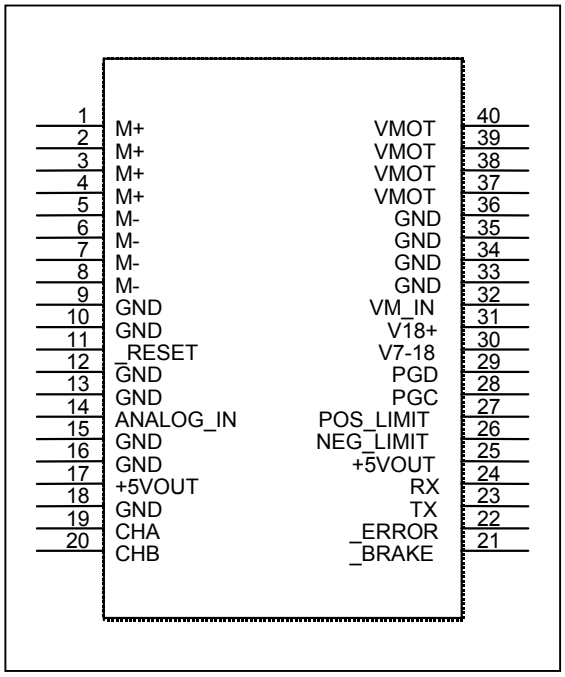
Date	Rev	Description	By
12-06-02	1	Original Implementation	L. Glazner
02-10-03	2	Fixed error in Read PID command description that included PID number as part of the response to this command.	L. Glazner

2. Introduction

MICRO PID Position Controller

FEATURES

- Up to 2A continuous current, up to 30VDC brushed motors
- Direct position control via 38.4KBPS, 19.2KBPS, or 9.6KBPS serial interface
- Stores up to 16 trapezoidal profile segments, and 3 PID settings
- Analog position control mode (1024 positions) with offset and multiplier functions
- Velocity control mode
- Built in data-logging modes for trapezoidal or direct control operating modes
- Brake, reset, negative limit switch, and positive limit switch inputs
- Custom software simplifies programming and testing
- X4 or X1 quadrature signal decoding



2.1 DESCRIPTION

The MICRO PID Position Controller is a complete PID position control module. When used with a brushed DC motor and a quadrature encoder, the MICRO PID Position Controller forms a highly versatile, low-cost, position control module, for low current servo motor control.

The MICRO PID Position Controller circuitry includes a 10-bit analog-to-digital conversion, indicator LEDs, quadrature decoder circuitry, and two channels of 10-bit PWM generation.

The MICRO PID Position Controller has four primary modes of operation. The first, direct position control mode allows a master unit to command the desired motor position up to 32 bits in length (two's compliment). In this mode of operation acceleration and maximum velocity are determined by the programmed PID filter values. In trapezoidal mode the MICRO PID Position Controller may be configured to run up to 16 consecutive trapezoidal movement segments. Each segment can be programmed with the desired position, acceleration limit, velocity limit, PID update rate, error-band, dwell time, and one of three user configurable PID filter settings. A master unit may initiate the trapezoidal movement segments, or the MICRO PID Position Controller may be programmed to implement the movements on power up.

Analog input (0V-5V, 1024 steps) control mode, with offset and multiplier settings can allow limited position control with just a potentiometer or other analog source. In Velocity control mode, with configurable rate of change and update rate settings, the MICRO PID Position Controller makes a useful platform for PID based motor speed control.

Additional features such as data logging (for direct and trapezoidal control modes) and user configurable limit switch settings round out this versatile and cost-effective position control solution. Support products including free programming/test software, the ICON Adapter Board (simplifies connectivity to a PC), and the 12V Active Cooling Kit make implementation of designs using these components painless and simple.

The unit measures 2.5" x 1.05" x 0.6". The package is a 40 pin DIP with 0.9" spacing between rows.

3. Engineering Specifications

3.1 Absolute Maximum Ratings

These are stress ratings only. Stresses above those listed below may cause permanent damage and/or affect device reliability. The operational ratings should be used to determine applicable ranges of operation.

Storage Temperature	-55°C to +150°C
Operating Temperature	0°C to +150°C
Motor Voltage (VM)	-0.3V to 30.0V
Voltage on logic control pins (J5)	-0.3V to +5.5V
Voltage on CHA, CHB, pins	-0.3V to +5.3V
Voltage on VM, M+, M-	35V transient spike
Motor Current Load	10A peak

3.2 DC Electrical Characteristics

At T_A = 25°C, V_{MOTOR} = 24V, I_{LOAD} = 0.5A F_{PWM} = 19.2kHz

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Logic Supply Voltage	+5VDC	4.5		5.5	V	
Logic Supply Current Source Capability	I5V		10	25	mA	5VOUT pins can supply a modest amount of current to external circuitry
Supply Current	IQ		20	50	mA	VM = 12V
Motor Voltage	VM	7		30	V	See section 3.6
ANALOG_IN voltage range	VANAIN	0		5	V	5V +/-4% is the full-scale input for the 10-bit ADC
ADC resolution	ADCRES	4.69	4.88	5.07	mV	Per ADC bit
CHA-B voltage	CHVOLT	-0.3		+5.3	V	CHA-B pins pulled to +5V with 10kΩ resistors
Peak load current	IPK			10	A	
Max current no cooling	INOCOOL		2		A	Tested at 12V 95% duty-cycle
Max current fan cooled (6.2CFM)	ICOOLED		2.75		A	Tested at 12V 95% duty-cycle
Intermittent current fan cooled (6.2CFM)	IINT		5		A	Tested at 12V 95% duty-cycle for 0.5S, 0% for 2S
Low Level Input RX pin	VRXIL			0.5	V	RX pin pulled to +5V with 10kΩ resistor
High Level Input RX pin	VRXIH	2.0			V	RX pin pulled to +5V with 10kΩ resistor
Low Level Input _BRAKE pin	VBRKIL			0.5	V	_BRAKE pin pulled to +5V with 10kΩ resistor
High Level Input _BRAKE pin	VBRKIH	2.0			V	_BRAKE pin pulled to +5V with 10kΩ resistor

note: “Typ” values are for design guidance only and are not guaranteed

DC Electrical Characteristics (continued)

At T_A = 25°C, V_M = 24V, I_{LOAD} = 0.5A F_{PWM} = 19.2kHz

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Low Level Input _RESET pin	V_{RSTIL}			0.5	V	_RESET pin pulled to +5V with 10kΩ resistor
High Level Input _RESET pin	V_{RSTIH}	2.0			V	_RESET pin pulled to +5V with 10kΩ resistor
Low Level Input CHA-B pins	V_{CHIL}			0.8	V	CHA-B pins pulled to +5V with 10kΩ resistors
High Level Input CHA-B pins	V_{CHIH}	3.5			V	CHA-B pins pulled to +5V with 10kΩ resistors
Low Level Input LIMIT switch pins	V_{LIMIL}			0.5	V	Limit switch pins pulled to +5V with 12.5kΩ res.
High Level Input LIMIT switch pins	V_{LIMIH}	2.0			V	Limit switch pins pulled to +5V with 12.5kΩ res.
Low Level Output TX pin	V_{TXOL}			0.6	V	TX pin pulled to +5V with 10kΩ resistor
High Level Output TX pin	V_{TXOH}	3.8		4.8	V	TX pin pulled to +5V with 10kΩ resistor
Maximum Analog Source Impedance	Z_{AIN}			2.5	kΩ	

note: "Typ" values are for design guidance only and are not guaranteed

3.3 AC Electrical Characteristics

At T_A = 25°C, V_M = 24V, I_{LOAD} = 0.5A F_{PWM} = 19.2kHz

Characteristic	Symbol	Min	Typ	Max	Unit	Notes
Communication bit period	T_{BIT}	26.0		104.2	uS	
Time for a command to be responded to	T_{TURN}	2		40	mS	
Time after power-up before device will communicate	T_{PWRUP}		1000	4000	mS	The onboard microcontroller blinks the P LED a number of times related to the operating mode (see section 6)
V_M rise time to ensure good reset	S_{VM}	0.05			V/mS	If this condition is not met then microcontroller may not power up correctly
PWM update rate	P_{PWM}	600		738	Updates/S	See figure 9
PWM Resolution	R_{PWM}		1024		bits	
PWM frequency	F_{PWM}	1.2	19.2	19.2	kHz	
Encoder Frequency	F_{ENC}		1.5	1.75	MHz	
Thermal shutdown		150	175	200	°C	
Thermal turn-on		120		170	°C	
Short-circuit current limit				20	A	
Over-current shutdown		6	8	10	A	Ratings for 150°C, 25°C, and –40°C

note: "Typ" values are for design guidance only and are not guaranteed

3.4 Mechanical Dimensions

The following diagram may be used to develop PCB carrying boards or enclosures used to fit the MICRO PID Position Controller into custom designs.

Figure 1: Mechanical Dimensions

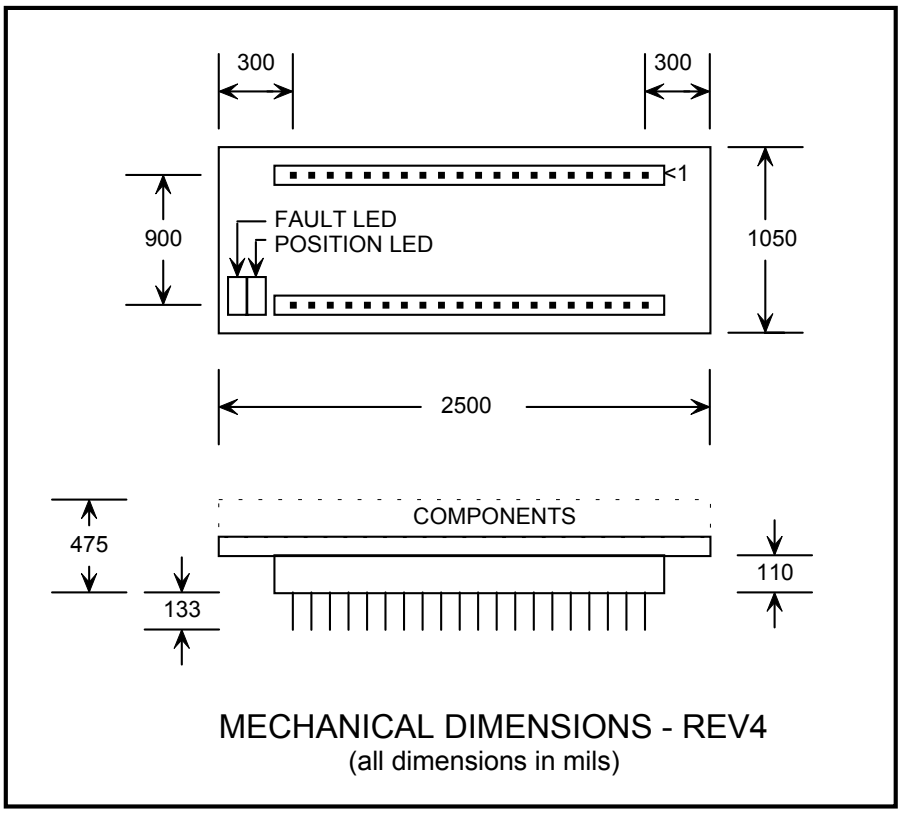


Figure 2: Mechanical Landmark Descriptions

Landmark	Type	Description
P-LED	LED – green	Position LED, lit when actual position read from incremental encoder is within error band of commanded position, also denotes operating mode on power up.
F-LED	LED – red	Error indicator this LED is lit when the MICRO PID Position Controller pulls the pin low. Error LED is lit for over-current or over-temperature faults.

3.5 Connectivity Overview

The DC motor and quadrature encoder connections are made through the 40 pin DIP connections. The individual pin designations are defined below.

Figure 3: MICRO PID Position Controller Pin Definitions

Pin	Name	Type	Description
1	M+	POWER	The positive lead of the brushed DC motor, or other load
2	M+	POWER	The positive lead of the brushed DC motor, or other load
3	M+	POWER	The positive lead of the brushed DC motor, or other load
4	M+	POWER	The positive lead of the brushed DC motor, or other load
5	M-	POWER	The negative lead of the brushed DC motor, or other load
6	M-	POWER	The negative lead of the brushed DC motor, or other load
7	M-	POWER	The negative lead of the brushed DC motor, or other load
8	M-	POWER	The negative lead of the brushed DC motor, or other load
9	GND	POWER	Ground return
10	GND	POWER	Ground return
11	_RESET	INPUT	Pulling _RESET low performs a hardware reset of the MICRO PID Position Controller, this pin should be left unconnected if not used
12	GND	POWER	Ground return
13	GND	POWER	Ground return
14	ANALOG_IN	INPUT	Analog control input, this pin should be left unconnected if the MICRO PID Position Controller is not operating in Analog mode
15	GND	POWER	Ground return
16	GND	POWER	Ground return
17	+5VOUT	POWER	+5VDC output, can supply up to 25mA, should be left unconnected if not needed
18	GROUND	POWER	Ground return
19	CHA	INPUT	Channel A input signal from quadrature incremental encoder, this input is pulled to +5VDC with a 10kΩ resistor, and is protected with a 5.6V zener diode
20	CHB	INPUT	Channel B input signal from quadrature incremental encoder, this input is pulled to +5VDC with a 10kΩ resistor, and is protected with a 5.6V zener diode
21	_BRAKE	INPUT	Pulling _BRAKE low forces the MICRO PID Position Controller to clear its PWM register and may reset many internal registers if in Trapezoidal mode
22	_ERROR	OUTPUT	The ERROR output is asserted (driven low) when the a fault occurs, the LED labeled “E” will be lit when _ERROR is asserted
23	TX	OUTPUT	TTL level, 8N1, serial transmission pin (data to the Master unit)
24	RX	INPUT	TTL level, 8N1, serial reception pin (data from the Master unit)
25	+5VOUT	POWER	+5VDC output, can supply up to 25mA, should be left unconnected if not needed
26	NEG_LIMIT	INPUT	Negative going limit switch, when asserted movement in the negative direction is prevented, logic level for assertion is software configurable, this input is pulled to +5VDC by a weak pull-up resistor
27	POS_LIMIT	INPUT	Positive going limit switch, when asserted movement in the positive direction is prevented, logic level for assertion is software configurable, this input is pulled to +5VDC by a weak pull-up resistor
28	NO_CONNECT	NC	This pin should be left unconnected
29	NO_CONNECT	NC	This pin should be left unconnected
30	V7-18	POWER	Connect this pin to VM_IN if your motor voltage is between +7 and +18VDC
31	V18+	POWER	Connect this pin to VM_IN if your motor voltage is greater than 18VDC
32	VM_IN	POWER	Filtered motor voltage connects to either pin 30 or 31
33	GND	POWER	Ground return
34	GND	POWER	Ground return
35	GND	POWER	Ground return
36	GND	POWER	Ground return
37	VM	POWER	The primary motor supply voltage connects to this pin (30V maximum)
38	VM	POWER	The primary motor supply voltage connects to this pin (30V maximum)
39	VM	POWER	The primary motor supply voltage connects to this pin (30V maximum)
40	VM	POWER	The primary motor supply voltage connects to this pin (30V maximum)

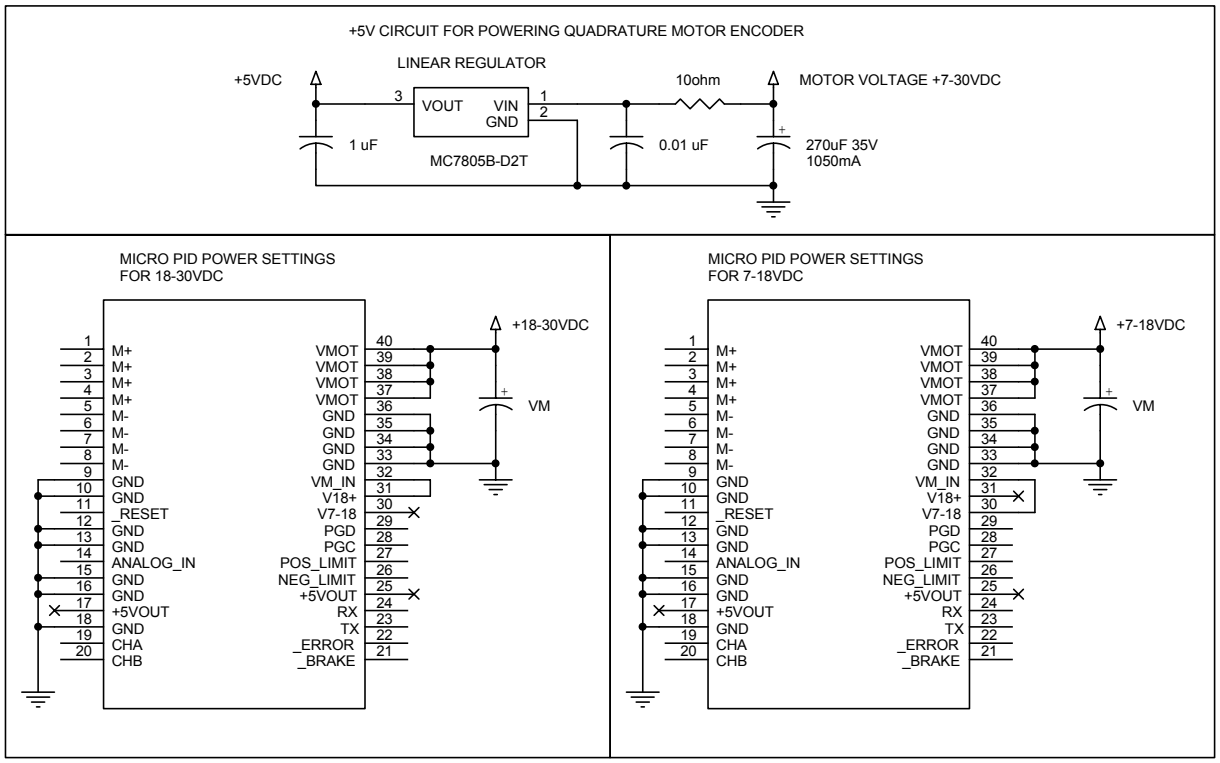
3.6 Power Supply Pin Connections

The MICRO PID Position Controller carries two linear regulators on board. The motor voltage at pins 37-40 is filtered and limited with a transient voltage suppressor (TVS). The filtered motor voltage is then fed out of the VM_IN pin (pin 32). To reduce the power dissipation in the linear regulators you must connect the filtered motor voltage to either the V7-18 pin (pin 30) or V18+ pin (pin 31).

For motor voltages of +18-30VDC connect the VM_IN pin to the V18+ pin (pin 31 to pin 32). For motor voltages of +7-18VDC connect the VM_IN pin to the V7-18 pin (pin 30 to pin 32).

The +5VOUT pins on the MICRO PID Position Controller can provide a limited amount of current to external circuits (about 25mA). In general it cannot provide enough current to power the quadrature encoder attached to your motor. These encoders typically require 50-250mA at 5V. Therefore you should expect to provide a separate +5V supply to your encoder's 5V input. A linear regulator like the one shown below can provide the current required. It is very important to add the VM capacitor (shown as a 270uF capacitor below) to provide instantaneous to the motor, so current is not drawn from the 0.01uF filter capacitor through the 10Ω resistor.

Figure 4: Power Supply Pin Connections

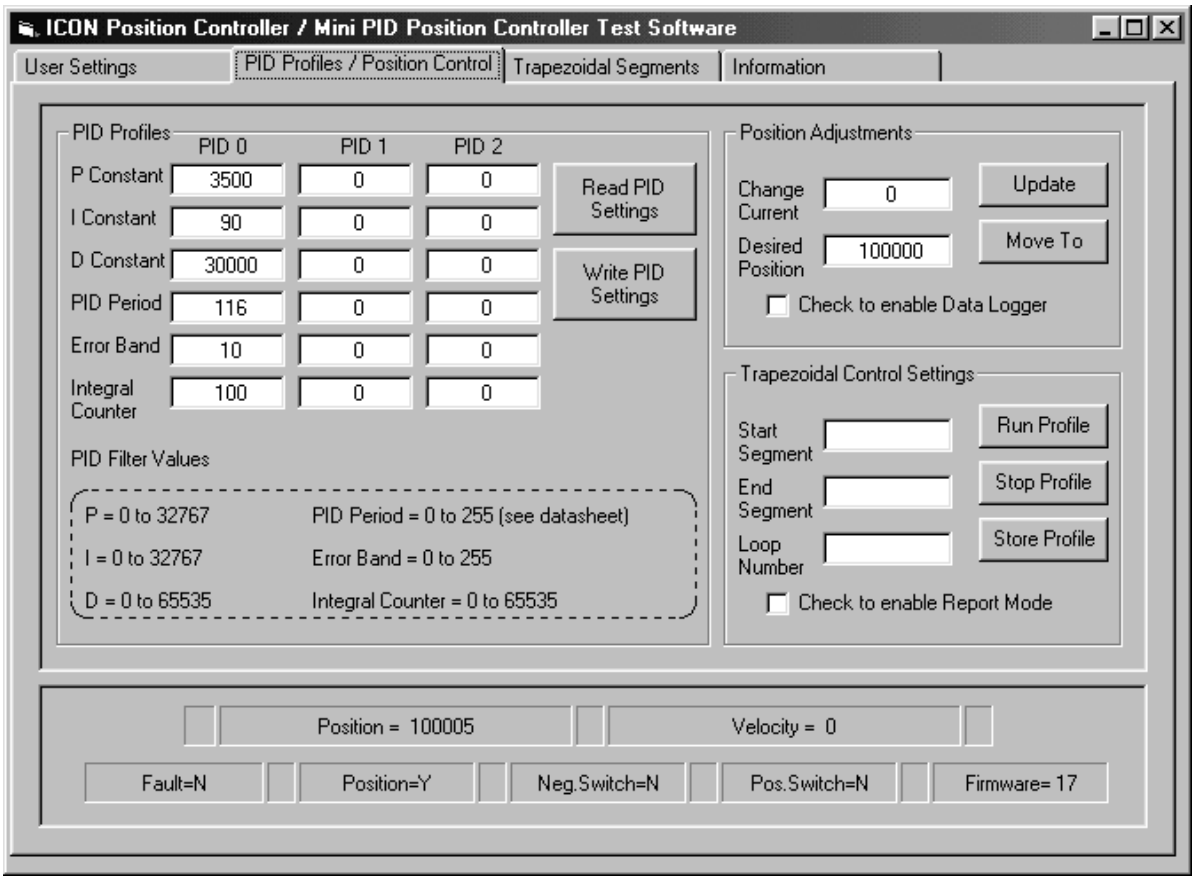


3.7 PID Filter Configuration and Settings

The MICRO PID Position Controller makes use of a proportional ("P"), integral ("I"), and derivative ("D"), or PID, filtering technique to effect changes in its output PWM signal relative to the error signal generated by the quadrature encoder connected CHA and CHB pins. The user inputs a desired position and an error signal is generated based on the difference between the desired position and the actual position.

The MICRO PID Position Controller is capable of storing 3 different PID settings. Three different PID settings can be useful when the device is operated in Trapezoidal mode and both large and small position moves are required. Both the Direct and Trapezoidal control modes have built in data logging functions that can be useful in tuning the PID algorithm for a particular system or series of movements. Solutions Cubed provides software for accessing all of the MICRO PID Position Controller functions, including some data logging when used in conjunction with a spreadsheet program. The PID filter programming window of the Solutions Cubed software is displayed below.

Figure 5: PID Programming Window for Solutions Cubed Software



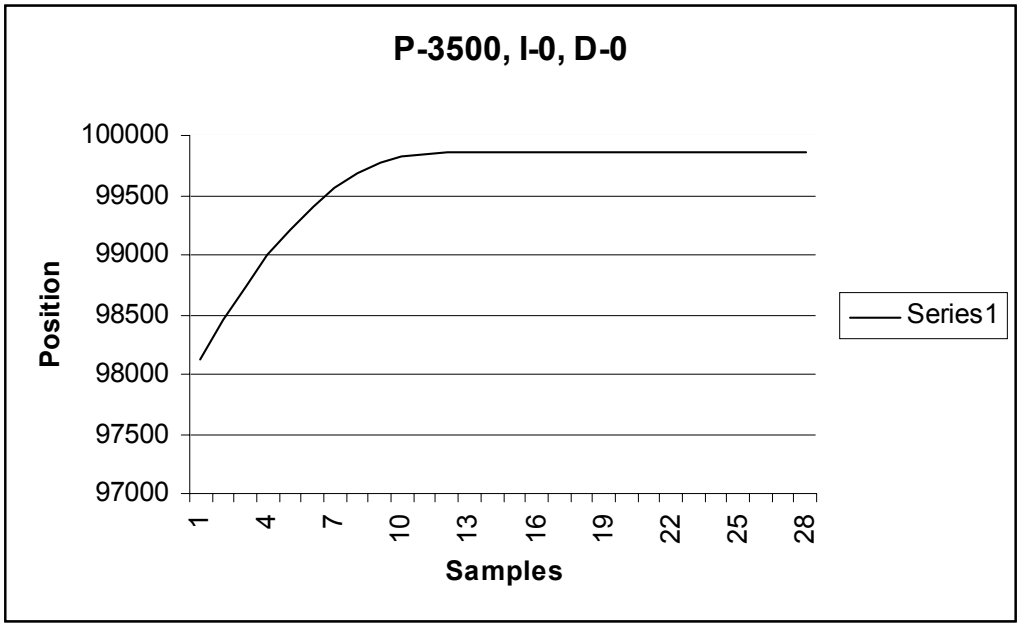
In the following section various charts are depicted showing a position step function (0 to +100,000) of a Pittman series 14000 LO_COG brush commutated DC motor, with a 500 counts-per-revolution Agilent 91X0 quadrature encoder. In each chart a move to position +100,000 is attempted. The graphs were generated using the Step Logger function in Direct position control mode.

3.7.1 Proportional Gain (can range from 0 to 32767)

The proportional aspect of the filter is simply the position error multiplied by the proportional constant. The "P" portion of the PID is responsible for the gross movement of the motor. Using proportional gain solely will typically result in a control response that undershoots or overshoots the desired position. As the actual position approaches the desired position the error signal becomes too small for the proportional gain to generate a significant PWM output, and therefore continue moving the motor.

Consider the graph below of a move to position +100,000. With "I" and "D" set to zero in the control algorithm, the actual position approaches the desired position and then stalls out. There needs to be some function that can act on the small remaining error to nudge the motor into the correct position.

Figure 6: Step Function – "P" Only

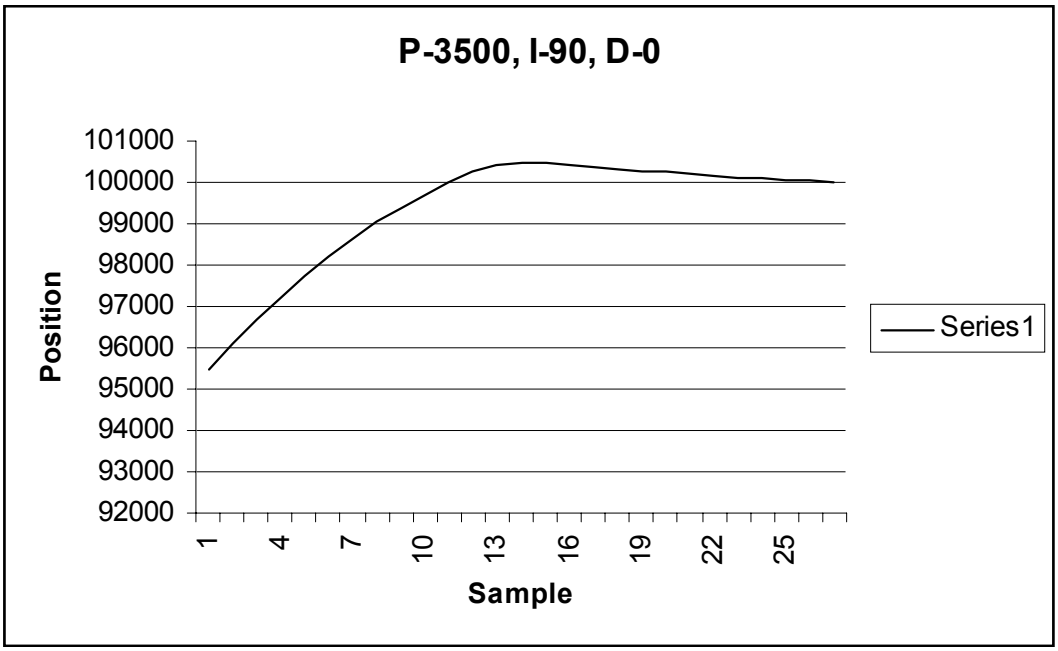


3.7.2 Integral Gain (can range from 0 to 32767)

The integral aspect of the filter is a continued summation of the position error multiplied by the integral constant. Several other factors effect the integral gain. First, if the PWM output reaches a maximum of 1023 then the output of the PID filter is considered "saturated". When the output is "saturated" the integral portion of the PID filter is skipped until the PWM output is reduced below 1023. This prevents "integral wind up" which has the effect of swamping out the "P" portion of the filter. In addition, the PID filter has an Integral Clear Counter value associated with it (see the section below for a detailed description of this counter). When this counter equals 0 the integral error stored thus far is reset to 0. Periodically resetting the stored integral error prevents small errors from building up over long periods of time and causing oscillations. The "I" portion of the PID is used to amplify small errors over time and can nudge a motor into the desired position. Overly large "I" settings will cause oscillations to occur around the desired position. The "I" setting is typically very small.

In the example below the inclusion of the "I" portion of the PID had two effects. First, it caused the motor to overshoot the desired position of +100,000. Second, it was able to correct for the overshoot by bringing the motor position back to the commanded position. Therefore the "I" term has the effect of increasing overshoot and decreasing the steady-state position error.

Figure 7: Step Function – "P" And "I"



3.7.3 Derivative Gain (can range from 0 to 65535)

The derivative gain constant is multiplied by the difference between the current position error (desired position – actual position) and a previous position error. In many PID algorithms the last position error is used (derivative term = derivative gain X (current error – last error)), but the MICRO PID Position Controller slightly modifies this method to increase the effectiveness of the "D" term of the PID algorithm. The actual algorithm used for the "D" term is (derivative term = derivative gain X (error $_{(N)}$ – error $_{(N-15)}$)). Adding the "D" term will reduce overshoot and increase rise time with respect to a step function. The "D" term will only impact operation of the motor controller when the motor is in motion. **When operating in Trapezoidal mode the derivative gain constant can, and probably should be, significantly reduced**. The velocity control algorithm used in Trapezoidal mode has an effect on overshoot similar to the effect of the derivative term of the PID filter. In fact a large "D" term used with Trapezoidal Mode can cause velocity spikes after the desired position has been reached.

3.7.4 PID Period (accepts specific values from 0 to 255)

The PID Period register is used to specify the number of PID updates that will occur during a 1-second interval. The PID Period register also defines the time-base used to measure a motor's velocity. The velocity settings applied by the user for Trapezoidal and Velocity control modes is defined as the number of pulses received during the time period specified by the PID Period register.

There are 4 update settings that are associated with a PWM frequency of 19.2KHz operation, 13 associated with 4.8KHz, and 16 associated with 1.2KHz. Most motors should be operated at 19.2KHz. The two lower PWM frequencies will cause audible whining and will not efficiently transfer power to motors.

Figure 8: Step Function – "P", "I", And "D"

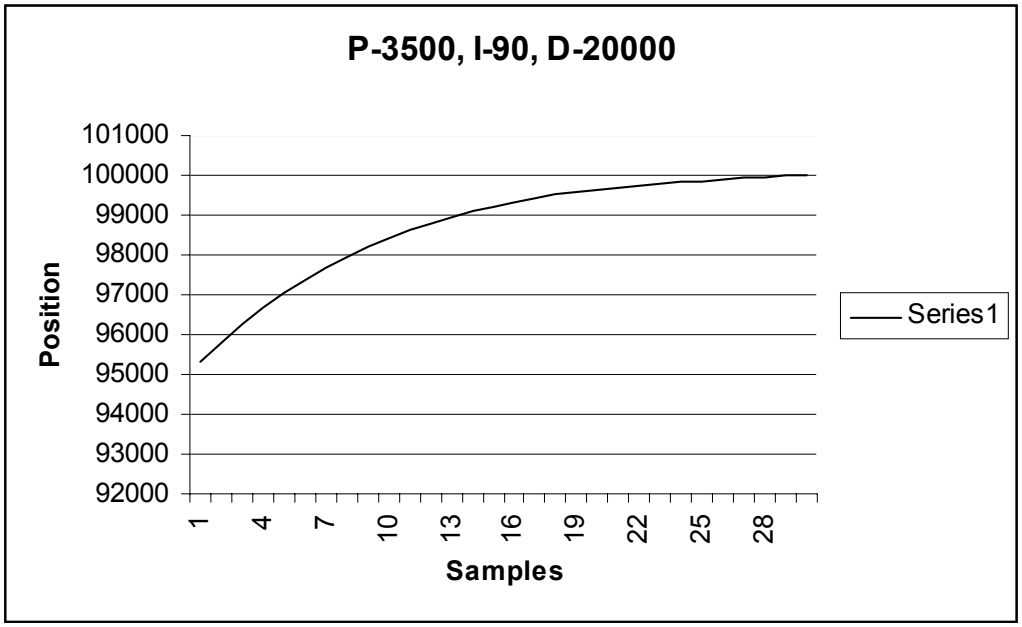


Figure 9: PID Period Constants

PID Period Constant	PID Period (Seconds)	Updates Per Second	PWM Frequency
100	1.354E-03	738	19.2KHz
108	1.459E-03	686	19.2KHz
116	1.563E-03	640	19.2KHz
124	1.667E-03	600	19.2KHz
29	1.667E-03	600	4.8KHz
37	2.084E-03	480	4.8KHz
45	2.500E-03	400	4.8KHz
53	2.917E-03	343	4.8KHz
61	3.334E-03	300	4.8KHz
69	3.750E-03	267	4.8KHz
77	4.167E-03	240	4.8KHz
85	4.584E-03	218	4.8KHz
93	5.001E-03	200	4.8KHz
101	5.417E-03	185	4.8KHz
109	5.834E-03	171	4.8KHz
117	6.251E-03	160	4.8KHz
125	6.668E-03	150	4.8KHz
7	1.667E-03	600	1.2KHz
15	3.334E-03	300	1.2KHz
23	5.001E-03	200	1.2KHz
31	6.668E-03	150	1.2KHz
39	8.334E-03	120	1.2KHz
47	1.000E-02	100	1.2KHz
55	1.167E-02	86	1.2KHz
63	1.334E-02	75	1.2KHz
71	1.500E-02	67	1.2KHz
79	1.667E-02	60	1.2KHz
87	1.834E-02	55	1.2KHz
95	2.000E-02	50	1.2KHz
103	2.167E-02	46	1.2KHz
111	2.334E-02	43	1.2KHz
119	2.500E-02	40	1.2KHz
127	2.667E-02	37	1.2KHz

3.7.5 Error Band (accepts values from 0 to 255)

The Error Band defines a band of positions around the commanded position that is considered by the MICRO PID Position Controller to be the equivalent of being "in position". For example if the Error Band is set to 10 and the desired position is +100,000, then any position between +99,990 and +100,010 will cause the MICRO PID Position Controller to generate a position error of "0".

3.7.6 Integral Clear Counter (accepts values from 0 to 65535)

The Integral Clear Counter is a setting that determines the number of PID periods that must elapse with no position error before the stored integral error is reset to 0. The position error must remain within the error band settings for all of the PID periods in order for the stored integral error to be reset. If the position error is ever out of the error band then the Integral Clear Counter is reset and the stored integral error is not cleared.

The overall effect of this setting is to clear the stored integral error before every new movement. The setting should be low enough to allow the stored integral error to be cleared before each movement, and in relation to the trapezoidal mode of operation the Integral Clear Counter should be about $1/10^{th}$ of Dwell Time setting.

3.8 PID Filter Tuning

Tuning the PID filter is typically a process of trial and error. The PID filter tuning process is dependent on your mechanical system, the electrical characteristics of your motor, and it's quadrature encoder. Much of the trial and error will be based on the allowable position error, overshoot, and response time, required by your mechanical system. PID filter tuning can be greatly simplified with software available at www.solutions-cubed.com, a PC, and the ICON Adapter Board (for connecting a PC to the MICRO PID Position Controller).

The PID filter tuning process described here is most useful when applied to step functions implemented while in Direct position control mode. First determine the minimum and maximum movement distances that you expect to be traveling. For example, if you have a 500CPR encoder on a DC motor, and you expect to be moving it anywhere from 2 revolutions to 100 revolutions, then your range of movement is between 1000 and 50,000. Select your step function to move from 0 to 25,000 (or one half the maximum distance you expect to move). Set your "I" and "D" constants to 0. Repeatedly adjust your "P" constant and perform the step movement from 0 to 25,000 until the motor ends up within a few hundred counts of the desired position. Then adjust the "I" value until the motor position moves within the error band. The "D" value can initially be set at 4X the "P" value (1X for Trapezoidal mode). The "P", "I", and "D" constants can then be modified (increased or decreased) until the commanded position is reached with a minimum of overshoot. Test step movements from the Minimum movement expected through the maximum. Some motors will behave differently in reverse so be sure to test your system with step functions going in both positive and negative directions.

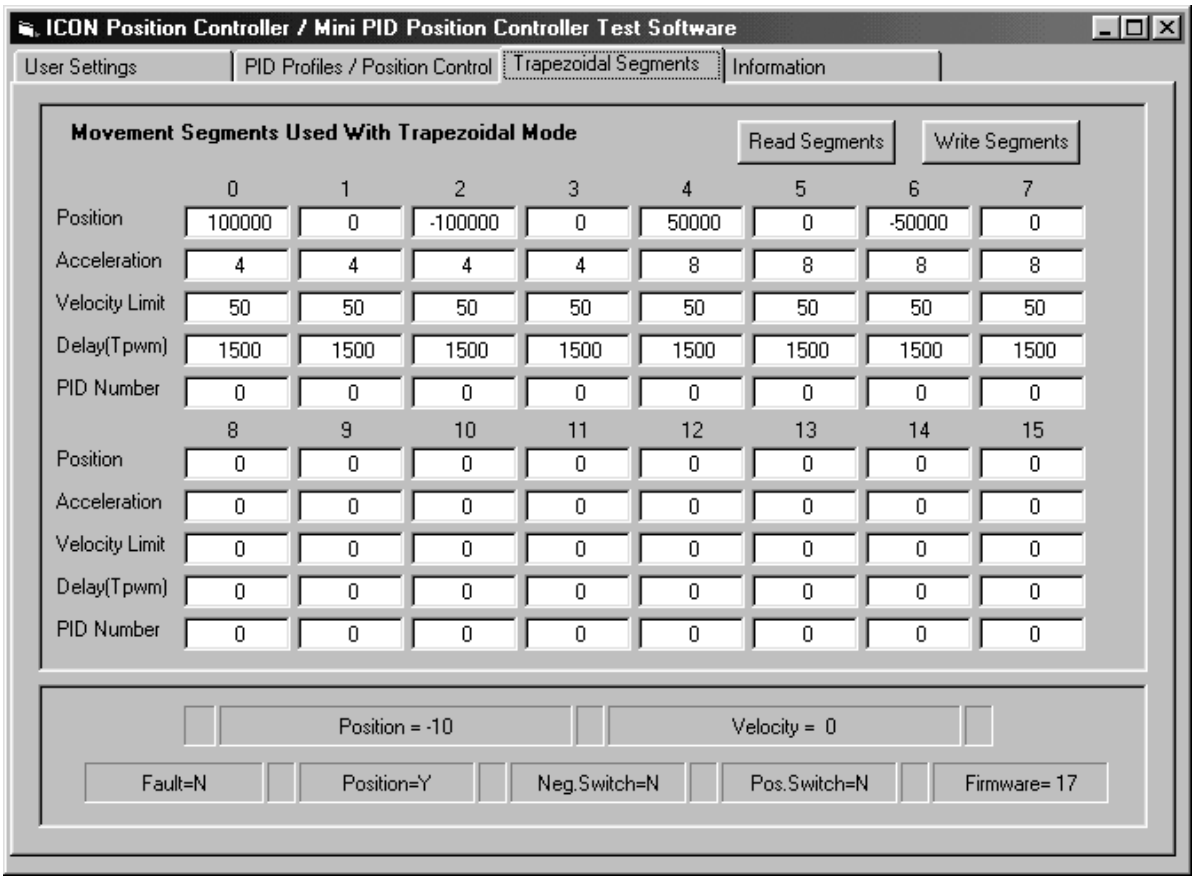
As a general rule your system will perform best if the position errors are large for even small motor movements. This means that for precise-small movements you should use a combination of high-resolution quadrature encoders and gear-reduced DC motors. Another weapon in your arsenal is the 4X_1X function built into the MICRO PID Position Controller. In 4X mode the output of your motor's quadrature encoder in resolved into 4 pulses for every 1 pulse received from the quadrature encoder. Using 4X mode can enhance the performance of your position controller with respect to small movements.

3.9 Trapezoidal Movement Profiles

Many mechanical systems are averse to abrupt starts and stops. For these systems the Trapezoidal mode of operation can be used to move from one position to another smoothly. A trapezoidal movement profile is made up of trapezoidal movement segments. Up to 16 segments may be stored

on board the MICRO PID Position Controller. Each segment may make use of 1 of 3 PID filter settings (see figure 5 for an example of the PID filter programming window).

Figure 10: Trapezoidal Profile Programming Window for Solutions Cubed Software


ICON Position Controller / Mini PID Position Controller Test Software
User Settings
PID Profiles / Position Control
Trapezoidal Segments
Information
Movement Segments Used With Trapezoidal Mode
Read Segments
Write Segments
0 1 2 3 4 5 6 7
Position 100000 0 -100000 0 50000 0 -50000 0
Acceleration 4 4 4 4 8 8 8 8
Velocity Limit 50 50 50 50 50 50 50 50
Delay(Tpwm) 1500 1500 1500 1500 1500 1500 1500 1500
PID Number 0 0 0 0 0 0 0 0
8 9 10 11 12 13 14 15
Position 0 0 0 0 0 0 0 0
Acceleration 0 0 0 0 0 0 0 0
Velocity Limit 0 0 0 0 0 0 0 0
Delay(Tpwm) 0 0 0 0 0 0 0 0
PID Number 0 0 0 0 0 0 0 0
Position = -10
Velocity = 0
Fault=N
Position=Y
Neg.Switch=N
Pos.Switch=N
Firmware= 17

The Start Segment (0-15), End Segment (0-15), and Number of Loops (1-255, with 0 used to denote infinite looping) define each trapezoidal movement profile. A movement profile will execute the Start Segment through the End Segment sequentially, and then repeat the segments the number of times defined in the Number of Loops setting. The trapezoidal movement profile is executed by the **Run Profile** command, and is effected by the **Store Profile** and **Stop Profile** commands.

Each trapezoidal segment consists of the commanded position, acceleration limit, velocity limit, dwell time (in PID periods), and the PID number to use with the defined trapezoidal segment.

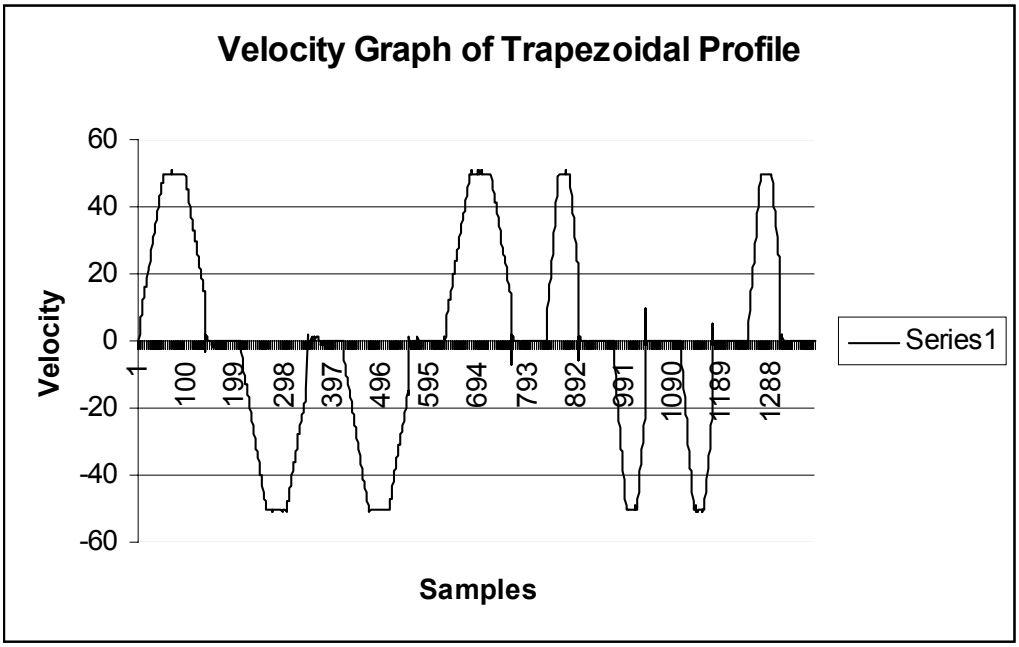
Figure 11: Components of a Trapezoidal Segment

Segment Component	Value Range	Description
Position	-2,147,483,648 to -2,147,483,647	two's compliment 32 bit number defines the position that the trapezoidal segment will move to
Acceleration	1 to 32,767	the maximum change in motor velocity while increasing motor speed
Velocity	1 to 32,767	The absolute value of the maximum motor velocity
Dwell Time	1 to 16,777,215	number of PID periods allowed for device to hunt for final position after movement exceeds commanded position
PID Number	0 to 2	PID filter settings to be used with this trapezoidal segment

A trapezoidal movement segment will attempt to smoothly accelerate the motor ("takeoff") up to a maximum velocity and then at the appropriate time decelerate the motor and ideally stop at the commanded position ("landing"). The rate of deceleration will approximate the rate of acceleration, but mechanical drag and the PID settings can cause the MICRO PID Position Controller to "land" early (decelerate to 0 before reaching the desired position) or late (reach the desired position before decelerating to 0). In the case of landing early the motor is not allowed to run at a speed less than the value loaded as the acceleration limit. This will cause the MICRO PID Position Controller to "crawl" to the final position at a low rate of speed. In the case of a late landing, the motor speed will be set to 0 as the desired position is reached. This can cause the motor to jerk to a stop.

Modifying the PID settings is one method of adjusting the landing of a trapezoidal segment. Another method of fine tuning the landing is by adjusting the maximum velocity or the acceleration limit settings. An excellent way to determine the smoothness of movement of your trapezoidal profile is to graph the velocity of the profile.

Figure 12: Velocity Graph of Trapezoidal Profile

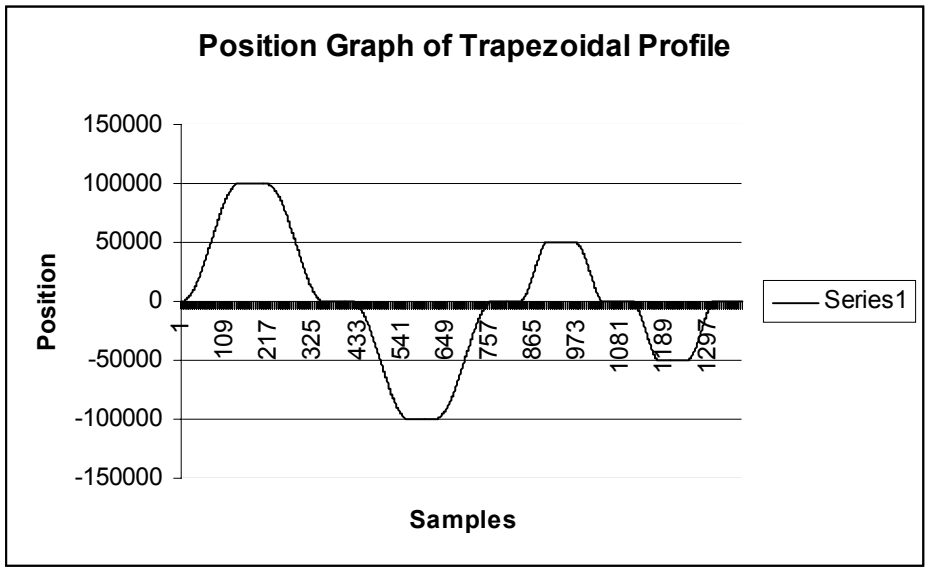


Several techniques may be used to adjust the "landing" of a particular trapezoidal segment. Figure 13 describes the two problems a trapezoidal movement segment may face and the effects of modifying the velocity and proportional gain settings to remedy the problem. **Reducing the "D" term of the PID filter can also remove velocity spikes that can occur at the end of a trapezoidal movement segment.**

Figure 13: Trapezoidal "Landing" Adjustments

Problem	Velocity Setting	Proportional Constant
"landing" overshoots desired position	decrease velocity maximum setting	Increase proportional gain
"landing" ends short of desired position	increase velocity maximum setting	Decrease proportional gain

Figure 14: Position Graph of Trapezoidal Profile



3.10 Step Logger Mode and Trap Logger Mode

Both the Direct and Trapezoidal control modes have functions built into them that allow a master unit to track position, velocity, and status information. Setting specific bits in the USER0 register accesses the data logging functions provided in the MICRO PID Position Controller (see the communication protocol section for detailed information on the format of the data returned).

3.10.1 Step Logger Mode

Step Logger mode is associated with Direct position control mode where the desired position is sent to the MICRO PID Position Controller by a master unit via the serial interface. Only the PID filter is used to determine motor speed with respect to position when operating in Direct control mode. The data will begin being sent when a **Write Desired Position** command is sent to the MICRO PID Position Controller. In Step Logger mode a block of data is sent to the master unit every 20th PID period. The block of data consists of the current position, velocity, and the contents of the STATUS register. When the Integral Clear Counter clears the stored integral error (see section 3.7.6) the Step Logger mode is also disabled. This allows the Step Logger mode to be used to accurately graph data associated with step function movements such as those displayed in figures 6, 7, and 8.

3.10.2 Trap Logger Mode

Trap Logger mode is associated with Trapezoidal position control mode where the desired trapezoidal segment is stored in the MICRO PID Position Controller by a master unit via the serial interface. The PID filter and the other settings stored in the trapezoidal segment are both used to determine motor speed with respect to position when operating in Trapezoidal control mode. The data will begin being sent when a **Run Profile** command is sent to the MICRO PID Position Controller. In Trap Logger mode a block of data is sent to the master unit every 20th PID period. The block of data consists of the current position, velocity, and the contents of the STATUS register. When the trapezoidal profile (a series of trapezoidal segments) is completed the Trap Logger mode is also disabled. This allows the Trap Logger mode to be used to accurately graph data associated with trapezoidal profiles such as those displayed in figures 12, and 14.

3.10.3 Logging Data in Velocity and Analog Modes

Although there are no built in functions for logging position or velocity data when operating in Velocity or Analog control modes the **Read Position** command may be issued by a master unit and the returned data may be used to graph the MICRO PID Position Controllers capabilities.

3.11 4X_1X Mode

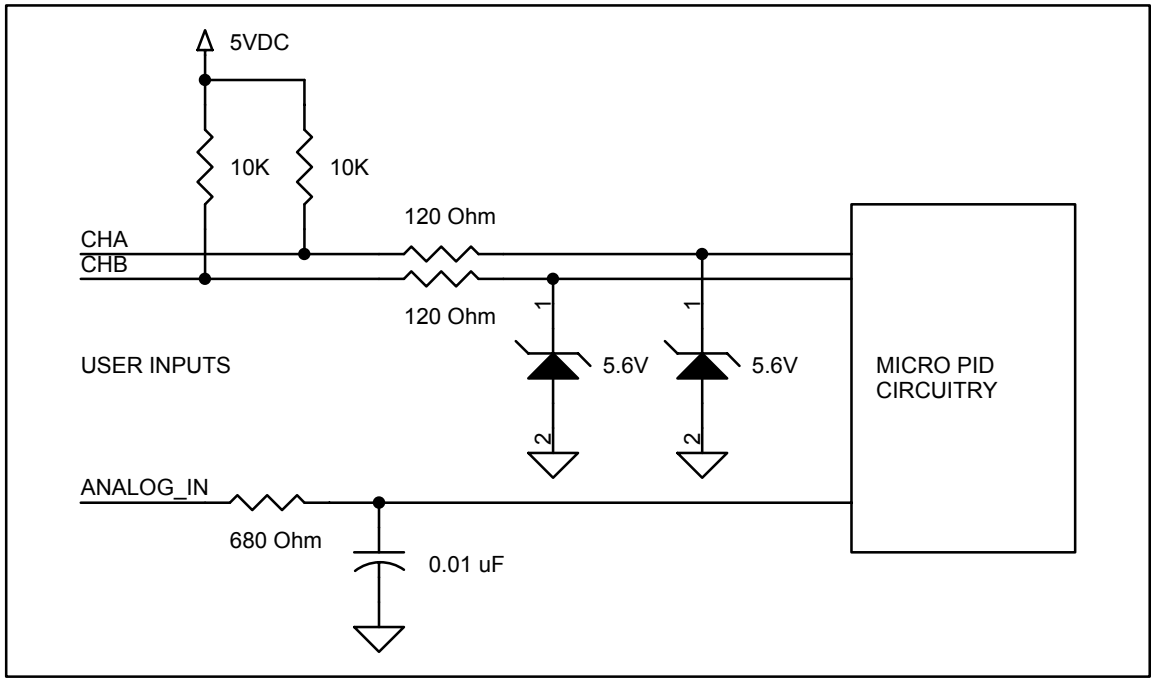
The user may select 4X or 1X decoding of the quadrature encoder signals by setting or clearing a bit in the USER0 register (the USER0 and USER1 registers are modified with the **Write User Registers** command). In 4X mode each set of signals from the quadrature encoder is decoded into 4 pulses. In 1X mode each set of signals from the quadrature encoder is decoded into 1 pulse. Therefore a 500CPR encoder could effectively be used as a 2000CPR encoder if the MICRO PID Position Controller was operating in 4X mode. Commanding a move to +100,000 while in 4X mode will result in a move of +25,000 of actual quadrature encoder counts. Likewise the velocity value measured will be four times greater in 4X mode as compared to the same velocity reading when operating in 1X mode.

4X mode may be used to increase resolution in position and velocity control. It can also be used to increase the position error signal allowing smaller movement distances to be implemented without using overly large PID constants.

3.12 ANALOG_IN, CHA, and CHB, Input Circuits

Some quadrature encoders or analog output circuits may be adversely effected by the protection and filtering circuits used by the MICRO PID Position Controller. The input protection circuits are detailed here for reference.

Figure 15: ANALOG_IN, CHA, and CHB Input Circuits



3.13 Two's Compliment Number System

The MICRO PID Position Controller can accept negative numbers for position and some velocity settings. In order to generate a two's compliment negative value take the binary or hexadecimal representation of the absolute value of the number, compliment it (every 1 becomes a 0 and every 0 becomes a 1) and add 1 to the result.

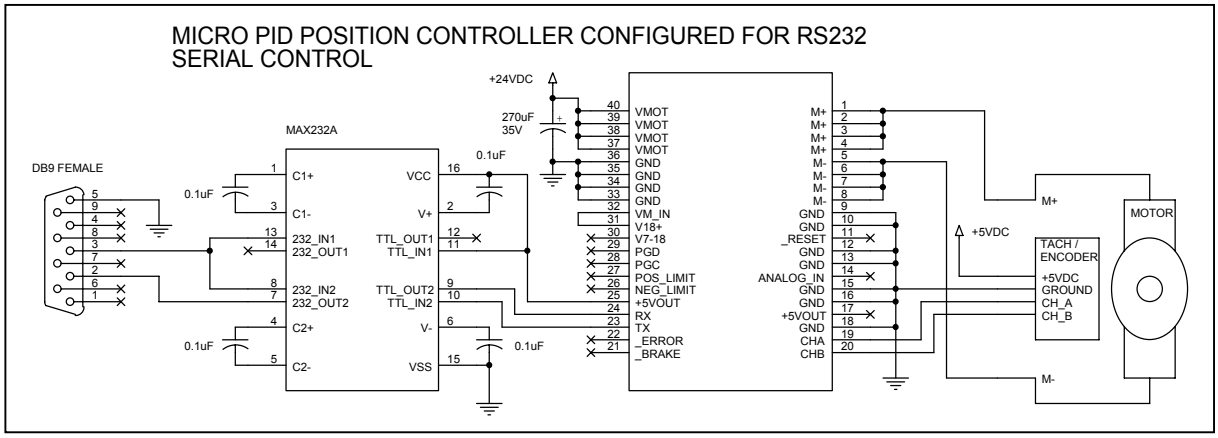
Figure 16: Two's Compliment Examples

Number	Absolute Value	Hexadecimal	Compliment	Two's Compliment
-1	1	0x00000001	0xFFFFFFFE	0xFFFFFFFF
-32768	32768	0x00008000	0xFFFF7FFF	0xFFFF8000
-100000	100000	0x000186A0	0xFFFE795F	0xFFFE7960

3.14 RS232 Conversion Circuitry

The primary method of programming and interfacing to the MICRO PID Position Controller is via a 38.4KBPS serial interface. The MICRO PID Position Controller supports TTL level serial communication (0V = logic "0" while +5V = logic "1"). Personal computer serial ports and many other serial devices operate on RS232 electrical specifications. RS232 serial data is transmitted under the same logic principles (therefore the data format does not change) but in order to increase noise immunity the RS232 electrical specification uses voltages from +10V(a logic "0") to -10V (a logic "1"). Therefore in order send data to the MICRO PID Position Controller from a PC additional circuitry converting serial data from RS232 to TTL and back is required. This can be done easily with a single IC such as the MAX232 or HIN232.

Figure 17: RS232 <-> TTL Conversion



Interfacing to the MICRO PID Position Controller can be further simplified by downloading free test software at www.solutions-cubed.com.

3.15 Limit Switch Functionality

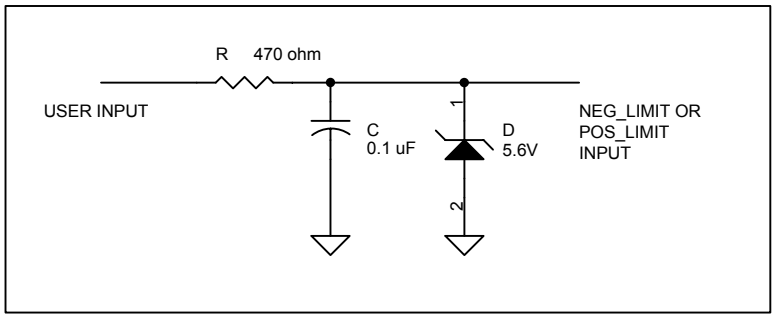
The MICRO PID Position Controller has two logic level inputs for limit switches. The state of assertion of these inputs (5V or 0V) is software configurable. The limit switches may be enabled or disabled by modifying bits 4 and 5 the USER1 register. The state of the limit switches (asserted or not asserted) can be read from bits 2 and 3 of the STATUS register.

The limit switch labeled NEG_LIMIT prevents movement in the negative direction when asserted. The limit switch labeled POS_LIMIT prevents movement in the positive direction when asserted. When a limit switch is asserted the controller will brake the motor until the velocity reading is "0", and will then take the actual position value and load it into the desired position register**. Communication with the controller is not possible while the motor is being braked.**

Asserting a limit switch in trapezoidal mode causes the next movement profile to end.

If the limit switch inputs are connected to contacts with long cables it may be necessary to protect the NEG_LIMIT and POS_LIMIT input pins from inductive voltage spikes. This may be the case if cabling exceeds 12". Figure 18 is one example of the kind of protective circuitry that can protect these input pins. Optoisolation may be necessary in extremely noisy environments. Always err on the side of caution.

Figure 18: Protection for Limit Switch Inputs



To modify limit switch functionality new settings can be written to the USER1 register with the **Write User Registers** command. A master unit can check the state of both limit switches by issuing a **Read User Registers** command and testing the contents of the STATUS register.

Figure 19: Registers and Bits Associated with the Limit Switches

Bit	USER1	STATUS	Description
0			
1			
2		NegSwitchFlag	1 = NEG_SWITCH is asserted
3		PosSwitchFlag	1 = POS_SWITCH is asserted
4	NegSWEnabled		1 = negative limit switch enabled
5	PosSWEnabled		1 = positive limit switch enabled
6	NegSWState		0 = asserted when input is 0V, 1 = asserted when input = 5V
7	PosSWState		0 = asserted when input is 0V, 1 = asserted when input = 5V

3.16 VM Capacitor

In some applications it may be necessary to provide additional capacitance between the VM and GND motor connections. This is typically true when you have long power supply leads, and your motor is drawing substantial current during stops or starts. If your MICRO PID Position Controller suffers from reset conditions, try connecting a large electrolytic capacitor (270uF+) across the VM and GND pins at the screw terminal connector. The capacitor should be have a ripple current rating of at least ½ of the peak load current.

3.17 Fault Conditions

The H-bridge on board the MICRO PID Position Controller has built in over-current and over-temperature fault conditions. This component has a flag output that denotes short-circuits, over-current, and over temperature conditions. The H-bridge flag is monitored by the MICRO PID Position Controller and is mirrored in the status.Fault flag.

The fault flag may be cleared with the **Clear Fault** command. Setting the user0.RetryFault flag to a "1" will cause the controller to clear the fault flags internally after 250 PID update periods have elapsed.

4. Operating Information

4.1 Overview

The MICRO PID Position Controller has four different modes of operation. The first of these modes is an analog position control mode, followed by trapezoidal position control mode, velocity control mode, and finally direct control mode. All of the user settings are stored in non-volatile memory. This allows the MICRO PID Position Controller to be used as a stand alone position controller. Or it may be used in conjunction with a master unit and controlled via the serial interface.

4.2 Analog Control Mode

Analog input signals should have a source resistance of less than 2.5kΩ and a range of 0-5V. When operating in Analog Control mode the MICRO PID Position Controller will monitor the analog input at the ANALOG_IN pin and convert the voltage to a 10-bit position setting. Therefore there can only be 1024 possible positions (0-1023). A value of 5V is the full-scale (highest) value that can be converted by the analog-to-digital converter (ADC). The resolution of the ADC is 4.88mV/step, accuracy will vary from system to system based on noise and off-board filtering techniques, but is typically accurate to +/-4%.

Two settings are associated specifically with the analog control mode. These are the Analog Offset and Analog Multiplier settings. Both of these settings can range from –32,768 to 32,767, and are modified by the **Write Analog Settings** command. The values currently used for the Analog Offset and Analog Multiplier can be read with the **Read Analog Settings** command. For the MICRO PID Position Controller to convert the analog input into a desired position setting requires 4 steps.

Step 1: Convert analog signal into digital value
Step 2: Add Analog Offset to digital result
Step 3: Multiply digital result by Analog Multiplier
Step 4: Divide digital result by 128

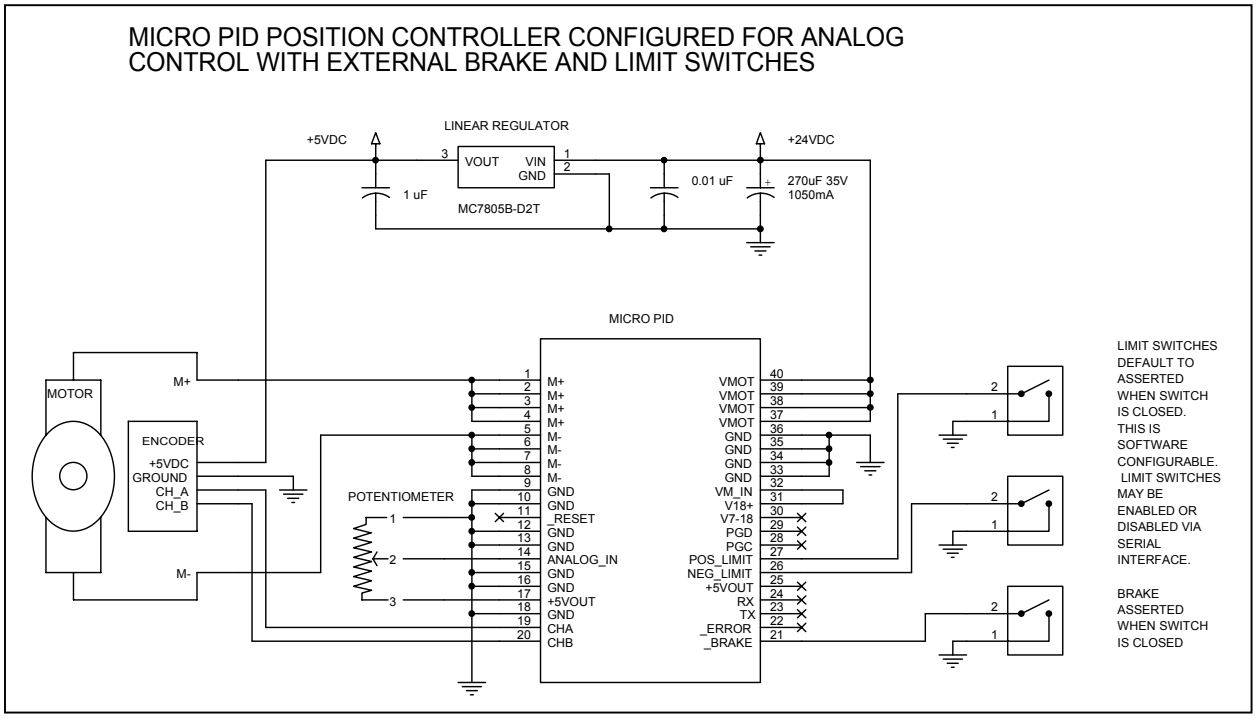
Example: Assume you have an analog input of 0-5VDC and a motor with a 500CPR quadrature encoder. You want to rotate the motor through a range of positions from –2,000 to +2,000 with a potentiometer. The 0 position is associated with 2.5V (therefore –2,000 at 0V and +2,000 at 5V) and any analog input greater than 5V will be treated as 5V.

Determining the Analog Multiplier is relatively easy. Take the span of positions you'll be using (-2,000 to 2,000 = 4,000) and divide that number by 8 (4,000/8 = 500). To calculate the Analog Offset take the span of the positions you'll be using and divide that number by 1,024 (4,000 / 1,024 = 3.9063). Then divide your lowest position by that number (-2,000 / 3.9063 = -512). So your Analog Multiplier is 500 (0x01F4 hexadecimal), and your Analog Offset is –512 (0xFE00 hexadecimal - 2's compliment).

Tips: Here are some tips to using analog position control with the MICRO PID Position Controller. First, you will only be able to achieve 1024 discrete positions. Therefore your resolution of position control is limited and the smoothness of operation degrades as the length of your move increases. Analog control works best with small moves, large proportional gain, and small derivative gain. Large proportional gains can cause jerky motion. Using geared motors and/or operating the MICRO PID Position Controller in 4X mode can help to smooth this jerkiness. You may have to make a tradeoff between system response time and smoothness of operation when operating in analog control mode. Use the free software located at www.solutions-cubed.com to program/test the MICRO PID Position Controller.

Setting bit 0 of the USER1 register with the **Write User Registers** command enables analog mode.

Figure 20: MICRO PID Position Controller Analog Mode Connections



4.3 Trapezoidal Control Mode

Trapezoidal control mode is used to generate trapezoidal shaped velocity profiles. Section 3.9 details trapezoidal movement segments. The MICRO PID Position Controller can store up to 16 separate "movement segments". Each movement segment consists of the distance to move, acceleration limit, velocity limit, a dwell time setting, and the PID filter to use for the movement. Each movement segment must be written to the MICRO PID Position Controller using the **Write Segment** command, and can be read with the **Read Segment** command. Movement segments are automatically stored in non-volatile memory with the **Write Segment** command.

A trapezoidal movement profile is made up of a series of movement segments (the series may have from 1 to 16 segments). The movement profile can be executed with the **Run Profile** command, it may be stopped with the **Stop Profile** command, and the movement profile may be stored in non-volatile memory with the **Store Profile** command. Furthermore, the MICRO PID Position Controller may be programmed to run movement profiles on power-up by setting bit 1 or the USER0 register with the **Write User Registers** command. To program the MICRO PID Position Controller to operate in trapezoidal mode follow the following steps.

Normal Operation

Step-1: Program for trapezoidal mode with the **Write User Registers** command
Step-2: Use the **Write Segment** command to load movement segment data
Step-3: Execute the movement profile with the **Run Profile** command

Running a Movement Profile on Power Up

Step-1: Program for trapezoidal mode, and run movement profile on power up with the **Write User Registers** command
Step-2: Use the **Write Segment** command to load movement segment data
Step-3: Store the desired movement profile with **Store Profile** command

To simplify programming free software can be found at www.solutions-cubed.com. Using a desktop PC and either the ICON Adapter Board or the circuit detailed in figure 17 the free software can allow you to quickly program and test the MICRO PID Position Controller.

The elements of a movement segment are detailed in figure 21.

Figure 21: Movement Segment Components

Element	Range	#Bytes	Description
Segment Number	0 to 15	1	The segment number differentiates between the movement segments.
Desired Position	-2,147,483,648 to +2,147,483,647	4	The desired position is the location that a movement segment will move to, this is a 32 bit - 2's compliment number that may be positive or negative. A rollover may occur if the desired position is close to the extreme upper or lower limit and overshoot occurs.
Acceleration Limit	1 to 32,767	2	The acceleration limit is used to define the maximum change in velocity from one PWM period to the next.
Velocity Limit	1 to 32,767	2	The velocity limit defines the peak velocity that the controller will attempt to reach during the trapezoidal movement. The velocity can be calculated by multiplying the velocity setting by the number of PWM updates per second (see figure 9).
Dwell Time	1 to 16,777,215	3	When the desired position is reached the controller will delay executing the next movement segment by a period of time defined by the Dwell Time setting. The duration of the delay is defined by the dwell time value multiplied by the PWM period.
PID Number	0 to 2	1	The trapezoidal mode of operation can associate one of three possible PID filter settings with each movement segment. This can be useful in fine-tuning movement profiles consisting of both large and small movement segments.

The elements of a movement profile are detailed in figure 22.

Figure 22: Movement Profile Components

Element	Range	#Bytes	Description
Start Segment	0 to 15	1	The start segment is the movement segment executed first in the movement profile.
End Segment	0 to 15	1	The end segment is the last movement segment executed by the movement profile. To execute a single segment just load the start and end segment registers with the same value. Otherwise the end segment should always be greater than the start segment.
Number of Loops	0 to 255	1	The loop number defines the number of times that the start through end segments is executed for the movement profile. Loading "0" will cause the movement profile to repeat continuously.

Example: Assume that you want to run a motor from the 0 position to the +100,000 position, wait there for 2 seconds, and return to the 0 position. And you would like to limit the maximum motor speed to roughly 25kHz. And you want to run this movement profile 5 times. The movement profile is made up of two movement segments (0 to +100,000 and +100,000 to 0). The MICRO PID Position Controller is programmed to operate at 19.2kHz with 600 PWM updates/second. Loading a "1" into the velocity limit would limit the velocity to 600Hz (1 encoder pulse per period and 600 periods per second). To calculate the velocity limit setting for 25kHz simply divide 25kHz by the number of PID updates per second. 25,000/600 = 41.67 or 42. The acceleration limit was arbitrarily set to 5. The Dwell Time setting is based on the PWM update period (2seconds x 600 updates per second = 1200 updates).

Movement Segment Settings

Element	Segment 0	Segment 1
Segment Number	0	1
Desired Position	+100,000	0
Acceleration Limit	5	5
Velocity Limit	42	42
Dwell Time	1200	1200
PID Number	0	0

Movement Profile Settings

Element	
Start Segment	0
End Segment	1
Number of Loops	5

Tips: Read section 3.9 on trapezoidal movement profiles. Make use of the Trap Logger function to verify the performance of the movement profiles that you design. Maintain a reduced derivative gain setting to eliminate velocity spikes at the culmination of a movement. The MICRO PID Position Controller will probably need the Dwell Time setting to be high enough for the integral gain to push the motor to the desired position, so allow time for this to occur. Use the free software located at www.solutions-cubed.com to program/test the MICRO PID Position Controller.

4.4 Velocity Control Mode

The MICRO PID Position Controller can be used to maintain a PID controlled velocity setting. Velocity control is limited in resolution by the PWM update rate and the setting for the Velocity Update Rate. The desired velocity set by the user is equivalent to the number of pulses from the quadrature encoder that the user wishes to see every PWM period. For example, if your PWM update rate was 600 updates per second, and you set the velocity control for 10, you could expect the motor speed to be set for (600 updates per second x 10 pulses per update =) 6000 pulses per second. If your motor had a 500CPR encoder you would see 12 motor shaft revolutions per second.

Several settings can effect the velocity control capability of the MICRO PID Position Controller. First, the PID Period Constant (see figure 9) defines the time base used for velocity control. The PID Period Constant is programmed with the **Write PID Values** command. Second, the Velocity Update Rate can be used to slow down the time base. The Velocity Update Rate is modified with the **Write-Store Velocity Settings** or **Write Velocity Settings** commands. To calculate the frequency resolution of the velocity control mode simply take the number of PWM updates per second for the PID Period Constant you're using and divide that number by the Velocity Update Rate. The Velocity Update Rate is the number of PWM periods that the MICRO PID Position Controller will accumulate incoming pulses from the quadrature encoder before implementing the PID filter.

If the PID Period Constant were 124 your PID update rate would be 600 updates per second. If you set the Velocity Update Rate to 6, then the PID update rate would be 600 / 6 = 100 updates per second (this is the **velocity update period**), and your frequency resolution would be 100Hz. **Increasing the Velocity Update Rate improves the resolution of your frequency measurement but decreases response time**.

Figure 23: "Write Velocity Settings" Components

Element	Range	#Bytes	Description
Desired Velocity	-32,768 to +32,767	2	Desired Velocity is the number of pulses from the quadrature encoder that you want to see during the velocity update period. Velocity control is bi-directional with negative velocities being expressed as 2's compliment negative numbers.
Velocity Update Rate	1 to 32,767	2	The Velocity Update Rate is the number of PWM periods that the MICRO PID Position Controller will accumulate incoming pulses from the quadrature encoder before implementing the PID filter.
Acceleration Limit	1 to 32,767	2	The acceleration limit is used to define the maximum change in velocity from one velocity update period to the next.

Velocity Control mode may be entered by setting bit 2 of the USER1 register with the **Write User Registers** command. Other commands that can be used with Velocity Control mode are **Write-Store Velocity Settings, Write Velocity Settings,** and **Read Velocity Settings**. The **Write-Store Velocity Settings** command stores the velocity settings in non-volatile memory. This will temporarily slow your motor. To make velocity changes on the fly without slowing the motor use the **Write Velocity Settings** command. To program the MICRO PID Position Controller to operate in velocity control mode follow the following steps.

Velocity Control Operation

Step-1: Program for velocity mode with the **Write User Registers** command

Step-2: Use the **Write-Store Velocity Settings** command to load Desired Velocity, Velocity Update Rate, and Acceleration Limit. These settings will be used on power up.

Step-3: Use the **Write Velocity Settings** command to modify motor speeds without storing them in EEPROM. This command should be used to change motor speed on the fly.

Example: You've got a motor with a gear ratio of 218:1 and a 500CPR encoder. You want to the motor shaft to rotate at ¼ revolution per second. By setting the MICRO PID Position Controller for 4X mode you can increase the encoder output to 2000CPR. The desired velocity for ¼ shaft rotations per second is (218 x 2,000 / 4) = 109,000 pulses per second. With a PID Period Constant of 124 and 600 updates per second the Desired Velocity should be set for 182 if the Velocity Update Rate is 1 (182 x 600 / 1 = 109,200). With a Velocity Update Rate of 1 the expected motor speed is 109,200 pulses per second and the frequency resolution is 600Hz. You could improve the resolution by increasing the Velocity Update Rate to 4. Then setting you Desired Velocity to 727 your frequency calculation would change to (727 x 600 / 4 = 109,050) and the frequency resolution improves to 600 / 4 = 150Hz.

Tips: Velocity control is accomplished by taking the desired velocity and adding it to the desired position. The **Write-Store Velocity Settings** command writes velocity data to EEPROM, which takes roughly 40ms. During this time motor speed is not updated on time and the speed control will have glitches. This will cause the motor to slow before executing the new settings. **Do not use commands that access EEPROM while operating in Velocity control mode.** Use the **Write Velocity Settings** command to update motor speed settings without introducing glitches in motor velocity control.

4.5 Direct Control Mode

Direct Control allows a master unit to control motor position through the serial communication protocol. This is typically done with the **Write Desired Position** command. Additional commands that may be useful are the **Write Position** and **Read Position** commands. The **Write Desired Position** command sends the desired motor position to the MICRO PID Position Controller. The **Write Position** command overwrites the actual position setting maintained by the MICRO PID Position Controller. To program the MICRO PID Position Controller to operate in direct control mode follow these steps.

Direct Control Operation

Step-1: Program for direct mode with the **Write User Registers** command
Step-2: Use the **Write Desired Position** command to move the motor to the desired position.

Example: The MICRO PID Position Controller will power up with a motor position of '0' regardless of the actual position of the motor in your mechanical assembly. Using the **Write User Registers** command enable the negative and positive limit switches and have the limit switches connected to your mechanical assembly. Then make multiple, short, positive, movements until the positive limit switch is asserted. Limit switch status can be determined with the **Read User Registers** command. If the motor position related to the positive limit switch is known, then the **Write Position** command may be used to load that value into the controller. Future use of the **Write Desired Position** command should then accurately position the motor.

Tips: Long and short moves may be performed more accurately with different PID settings. If this is the case in your system you can load different PID settings with the **Write PID Values** command prior to sending different movement commands with the **Write Desired Position** command.

5. Communication Protocol

5.1 Overview

The MICRO PID Position Controller makes use of a serial communication protocol for programming and some of the interface methods. Generally the user will have to utilize the serial interface at least once before placing the MICRO PID Position Controller into operation. For instance, if the device was to be operated in Analog control mode the Analog Offset, Analog Multiplier, and the PID settings would need to be evaluated and modified via the serial interface before the MICRO PID Position Controller can be operated as a stand alone analog position controller. Accessing the various registers within the MICRO PID Position Controller can be simplified by making use of the custom software available at www.solutions-cubed.com. The serial communication protocol used by the MICRO PID Position Controller is TTL level, 8N1, 38.4KBPS, true data (see section 5.4 for other baud rates). Connecting to a PC serial port will require an RS232 converter (see figure 17). The checksum is a simple modulo 256 summation of all bytes in the message previous to the checksum. Message components greater than 1 byte in length are sent and received least significant byte first. For example, if sending 10,000 as two bytes the bytes sent would be first 0x10, then 0x27 (both in hexadecimal (10,000 decimal = 0x2710 hexadecimal)). **The commands in bold access EEPROM when executed.** It is best to implement these commands while the motor is stopped; this is especially true when operating in Velocity control mode.

5.2 Command Set

Figure 24: Position Controller Command Set

Command	Description
Write Segment	Stores movement segment data for segments 0-15 in EEPROM (single segment)
Read Segment	Reads movement segment data for segments 0-15(single segment)
Run Profile	Initiates a trapezoidal movement profile from start to end segment and 1 to 255 loops or continuous
Stop Profile	Stops a trapezoidal profile while it is being processed
Store Profile	Stores trapezoidal profile settings in EEPROM so they can be implemented on power up (when user0.RunFromEEPROM flag is set)
Write PID Values	Writes PID constant settings to EEPROM for PID 0, 1, or 2
Read PID Values	Read PID constant settings for PID 0, 1, or 2
Write Analog Settings	Writes the Analog Offset and Analog Multiplier values to EEPROM
Read Analog Settings	Reads current values for Analog Offset and Analog Multiplier
Write-Store Velocity Settings	Writes Desired Velocity, Velocity Update Rate, and Acceleration values and stores them in EEPROM
Write Velocity Settings	Writes Desired Velocity, Velocity Update Rate, and Acceleration values but **does not** store them in EEPROM
Read Velocity Settings	Reads current settings for Desired Velocity, Velocity Update Rate, and Acceleration
Write User Registers	Write to EEPROM the USER0 and USER1 flag registers
Read User Registers	Reads USER0, USER1, and STATUS flag registers
Write Address	Write to EEPROM a new address value
Write Position	Overwrites the current position value generated by the quadrature encoder, can be used to reset the position value or write a different position value
Read Position	Reads current position generated by quadrature encoder
Write Desired Position	Writes desired position value, can be considered a “Move To” command when operating in Direct Control mode.
Clear Fault Flag	Clears fault condition flag caused by over-temperature or over-current condition
Read Firmware Revision	Reads current firmware revision, can be compared to errata sheet document to track known firmware errors
Logging Mode Response	Details format of serial response received from Step Logger or Trap Logger data logging functions

5.2.1 Write Segment

Description: The Write Segment command sends the various components of a single trapezoidal movement segment (0 through 15) to the MICRO PID Position Controller. The segment data is accessed by the **Run Profile** command when the controller is operated in Trapezoidal Control mode.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	224	1	224 (0xE0)
Address	0 to 255	1	1 (0x01)
Segment Number	0 to 15	1	0 (0x00)
Desired Position	-2,147,483,648 to +2,147,483,647	4	10,000 (0x00002710)
Acceleration Limit	1 to 32,767	2	5 (0x0005)
Velocity Limit	1 to 32,767	2	50 (0x0032)
Dwell Time	0 to 16,777,215	3	1500 (0x0005DC)
PID Number	0 to 2	1	0 (0x00)
Checksum	0 to 255	1	48 (0x30)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: For the example shown here the serial data string in hexadecimal would appear as "E0, **01**, 00, **10, 27, 00, 00**, 05, 00, **32, 00,** DC, 05, 00, **00**, 30" with the command byte being received first by the MICRO PID Position Controller. Every other component of the message is shown in bold to help visually separate the various components of the command.

5.2.2 Read Segment

Description: The Read Segment command returns the values stored in EEPROM for a single trapezoidal movement segment with a value of 0 through 15.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	225	1	225 (0xE1)
Address	0 to 255	1	1 (0x01)
Segment Number	0 to 15	1	0 (0x00)
Checksum	0 to 255	1	226 (0xE2)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Header	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
Segment Number	0 to 15	1	0 (0x00)
Desired Position	-2,147,483,648 to +2,147,483,647	4	10,000 (0x00002710)
Acceleration Limit	1 to 32,767	2	5 (0x0005)
Velocity Limit	1 to 32,767	2	50 (0x0032)
Dwell Time	0 to 16,777,215	3	1500 (0x0005DC)
PID Number	0 to 2	1	0 (0x00)
Checksum	0 to 255	1	240 (0xF0)

Notes:

5.2.3 Run Profile

Description: Run Profile is used while operating in Trapezoidal Control mode. When this command is received each trapezoidal movement segment from the Start Segment through the End Segment is executed (the Start-End segment numbers relate to the "Segment Number" defined in the Write Segment and Read Segment commands). The movement segments can be executed continuously by setting the Number of Loops value to 0, or can be repeated 1 to 255 times by setting the Number of Loops value to a number ranging from 1 to 255. To execute a single trapezoidal segment set the Start Segment and End Segment to the same value.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	226	1	226 (0xE2)
Address	0 to 255	1	1 (0x01)
Start Segment	0 to 15	1	0 (0x00)
End Segment	0 to 15	1	2 (0x02)
Number of Loops	0 to 255	1	5 (0x05)
Checksum	0 to 255	1	234 (0xEA)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: The Number of Loops setting is the number of times the Trapezoidal Profile (Start Segment through End Segment) will be executed. In the example above segments 0,1, and 2 would be executed in that order 5 times. The movement profile would then end. If the Number of Loops setting is 0 then the Start through End segments are executed continuously.

5.2.4 Stop Profile

Description: The Stop Profile command is used to stop trapezoidal movement profiles while they are in progress. This command clears all of the internal flags used to track where the controller is at during a specific trapezoidal profile. When executed the actual motor position becomes the desired motor position.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	227	1	227 (0xE3)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	228 (0xE4)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes:

5.2.5 Store Profile

Description: The Store Profile command stores the trapezoidal movement profile in EEPROM.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	228	1	228 (0xE4)
Address	0 to 255	1	1 (0x01)
Start Segment	0 to 15	1	0 (0x00)
End Segment	0 to 15	1	2 (0x02)
Number of Loops	0 to 255	1	5 (0x05)
Checksum	0 to 255	1	236 (0xEC)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes:

5.2.6 Write PID Values

Description: The Write PID Values command is used to store the "P", "I", and "D" constants, and other PID filter values in EEPROM. Regardless of the PID number, if the controller is executing movements it uses the last PID settings written by this command. PID values should only be written when no movement is occurring. Trapezoidal movements will load the PID value associated with the Movement Segment it is executing, all other modes of operation make use of the last PID settings written to the device. On power up PID-0 values are loaded.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	229	1	229 (0xE5)
Address	0 to 255	1	1 (0x01)
PID Number	0 to 2	1	0 (0x00)
P Constant	0 to 32,767	2	12,000 (0x2EE0)
I Constant	0 to 32,767	2	125 (0x007D)
D Constant	0 to 65,535	2	35,000 (0x88B8)
PID Period	0 to 255	1	116 (0x74)
Error Band	0 to 255	1	10 (0x0A)
Integral Clear Counter	0 to 65,535	2	1000 (0x03E8)
Checksum	0 to 255	1	26 (0x1A)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes:

5.2.7 Read PID Values

Description: Reads PID filter settings for desired PID number from EEPROM

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	230	1	230 (0xE6)
Address	0 to 255	1	1 (0x01)
PID Number	0 to 2	1	0 (0x00)
Checksum	0 to 255	1	231 (0xE7)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
P Constant	0 to 32,767	2	12,000 (0x2EE0)
I Constant	0 to 32,767	2	125 (0x007D)
D Constant	0 to 65,535	2	35,000 (0x88B8)
PID Period	0 to 255	1	116 (0x74)
Error Band	0 to 255	1	10 (0x0A)
Integral Clear Counter	0 to 65,535	2	1000 (0x03E8)
Checksum	0 to 255	1	213 (0xD5)

Notes:

5.2.8 Write Analog Settings

Description: The Write Analog Settings is used to write offset and multiplier settings used to translate the analog measurement value (0-1023) to a specific desired position value.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	231	1	231 (0xE7)
Address	0 to 255	1	1 (0x01)
Analog Offset	-32,768 to 32,767	2	-512 (0xFE00)
Analog Multiplier	-32,768 to 32,767	2	1024 (0x0400)
Checksum	0 to 255	1	234 (0xEA)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: See section 4.2 for more information on the calculation used to transform the analog input to a digital position value.

5.2.9 Read Analog Settings

Description: The Read Analog Settings is used to read the current offset and multiplier settings stored in the controller.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	232	1	232 (0xE8)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	233 (0xE9)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
Analog Offset	-32,768 to 32,767	2	-512 (0xFE00)
Analog Multiplier	-32,768 to 32,767	2	1024 (0x0400)
Checksum	0 to 255	1	163 (0xA3)

Notes:

5.2.10 Write-Store Velocity Settings

Description: The Write-Store Velocity Settings is used to write velocity control settings to the controller and store those settings in EEPROM. This will cause the motor to slow before implementing the new settings. To change velocity control settings on the fly the Write Velocity Settings command should be used.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	233	1	233 (0xE9)
Address	0 to 255	1	1 (0x01)
Desired Velocity	-32,768 to 32,767	2	-100 (0xFF9C)
Velocity Update Rate	1 to 32,767	2	2 (0x0002)
Desired Acceleration	1 to 32,767	2	5 (0x0005)
Checksum	0 to 255	1	140 (0x8C)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: See section 4.4 for more information on the calculation used to determine the motor velocity and its relationship to the velocity settings.

5.2.11 Write Velocity Settings

Description: The Write Velocity Settings is used to write velocity control settings to the controller and **does not** store those settings in EEPROM. To change motor velocity settings on the fly the Write Velocity Settings command should be used.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	249	1	249 (0xF9)
Address	0 to 255	1	1 (0x01)
Desired Velocity	-32,768 to 32,767	2	-100 (0xFF9C)
Velocity Update Rate	1 to 32,767	2	2 (0x0002)
Desired Acceleration	1 to 32,767	2	5 (0x0005)
Checksum	0 to 255	1	156 (0x9C)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: See section 4.4 for more information on the calculation used to determine the motor velocity and its relationship to the velocity settings.

5.2.12 Read Velocity Settings

Description:

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	234	1	234 (0xEA)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	235 (0xEB)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
Desired Velocity	-32,768 to 32,767	2	-100 (0xFF9C)
Velocity Update Rate	1 to 32,767	2	2 (0x0002)
Desired Acceleration	1 to 32,767	2	5 (0x0005)
Checksum	0 to 255	1	67 (0x43)

Notes: See section 4.4 for more information on the calculation used to determine the motor velocity and its relationship to the velocity settings.

5.2.13 Write User Registers

Description: Write User Registers is used to define the operating mode of the controller. Various other functions such as software reset, restoration of default values, accessing data logger modes, and settings associated with the limit switches are controlled via the USER0 and USER1 registers.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	235	1	235 (0xEB)
Address	0 to 255	1	1 (0x01)
USER0	0 to 255	1	34 (0x22)
USER1	0 to 255	1	49 (0x31)
Checksum	0 to 255	1	156 (0x9C)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: See section 5.3 for USER0, USER1, and STATUS flag definitions.

5.2.14 Read User Registers

Description: Read User Registers is used to read the current operating mode of the controller. The STATUS register contains additional information pertaining to the limit switches, fault conditions, and whether or not the controller has placed the motor within the defined error-band .

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	236	1	236 (0xEC)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	237 (0xED)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
USER0	0 to 255	1	34 (0x22)
USER1	0 to 255	1	49 (0x31)
STATUS	0 to 255	1	2 (0x02)
Checksum	0 to 255	1	246 (0xF6)

Notes: See section 5.3 for USER0, USER1, and STATUS flag definitions.

5.2.15 Write Address

Description: The Write Address command stores the new address value EEPROM and replaces the existing device address with the new value. All future serial communication must use the new address value to be considered valid.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	237	1	237 (0xED)
Address	0 to 255	1	1 (0x01)
New Address	0 to 255	1	2 (0x02)
Checksum	0 to 255	1	240 (0xF0)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes:

5.2.16 Write Position

Description: The Write Position command is used to reset the current motor position to the value defined by the Position component of the command. This can be used to reset the motor position to 0, or to some other setting.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	238	1	238 (0xEE)
Address	0 to 255	1	1 (0x01)
Position	-2,147,483,648 to +2,147,483,647	4	0 (0x00000000)
Checksum	0 to 255	1	239 (0xEF)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: This command takes the desired position as well as the actual position and loads the position component of the command into these registers. For example, if the controller was sitting at position +100,000 and this command sent the value –100,000 the controller would then operate as if it were at position –100,000. If this command was executed while the motor was stopped there would be no movement generated.

5.2.17 Read Position

Description: The Read Position command reads the actual motor position at the time that the command is received.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	239	1	239 (0xEF)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	240 (0xF0)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
Position	-2,147,483,648 to +2,147,483,647	4	-10,000 (0xFFFFD8F0)
Velocity	-32,768 to +32,767	2	100(0x0064)
Checksum	0 to 255	1	103 (0xCB)

Notes:

5.2.18 Write Desired Position

Description: The Write Desired Position command is to generate “Move To” commands when operating in Direct Control mode. It can also effect motor movement in Analog, Velocity, or Trapezoidal operating modes, but would not typically be used with those modes of operation.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	240	1	240 (0xF0)
Address	0 to 255	1	1 (0x01)
Position	-2,147,483,648 to +2,147,483,647	4	+50,000 (0x0000C350)
Checksum	0 to 255	1	04 (0x04)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: It is important to specify the actual position versus the desired position. The actual position is the position the controller reads from the quadrature encoder connected to the motor itself. The desired position is the position the motor controller will try to attain. The error signal fed into the PID filter is the difference between the desired motor position and the actual motor position. Analog, Velocity, and Trapezoidal control modes generate the desired position settings internally.

5.2.19 Clear Fault Flag

Description: This command will clear the STATUS.Fault flag caused by over-current or over-temperature conditions. Care should be taken not to clear the fault flag if a fault condition still exists.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	241	1	241 (0xF1)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	242 (0xF2)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
ACK	6	1	6 (0x06)

Notes: The USER0.RetryFault setting (enabled by the Write User Registers command) will automatically clear the fault condition after 250 PID periods.

5.2.20 Read Firmware Revision

Description: The current firmware revision can be read with this command and may be used to reference errata documents at www.solutions-cubed.com to determine if an error exists in your firmware. Errors may or may not exist, and if they do exist may not effect your design.

Component of Command	Range of Values	Number of Bytes	Decimal(Hex) Example
Command	255	1	255 (0xFF)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	0 (0x00)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Response Byte	160	1	160 (0xA0)
Address	0 to 255	1	1 (0x01)
Firmware Revision	0 to 255	1	16 (0x10)
Checksum	0 to 255	1	177 (0xB1)

Notes:

5.2.21 Logging Mode Response

Description: The Step Logger and Trap Logger data logging functions accessed through the USER0 register, and used with Direct and Trapezoidal control modes respectively, output data in the format described below. Data is sent every 20 PID periods.

Component of Response	Range of Values	Number of Bytes	Decimal(Hex) Example
Response Byte	176	1	177 (0xB0)
Address	0 to 255	1	1 (0x01)
Actual Position	-2,147,483,648 to +2,147,483,647	4	+100,000 (0x00186A0)
Actual Velocity	-32,768 to +32,767	2	-53 (0xFFCB)
STATUS Byte	0 to 255	1	0 (0x00)
Checksum	0 to 255	1	162 (0xA2)

Notes: In Step Logger mode the data will cease being sent when the Integral Clear Counter counts to zero. Therefore the motor must be in position for the controller to exit Step Logger mode. Executing a **Write Position** command while in Step Logger mode also has the effect of ceasing the data logging function. Trap Logger mode ceases to send data when the numbers of loops in the Trapezoidal Movement Profile are completed.

5.3 USER0, USER1, and STATUS Flags

Configuring the MICRO PID Position Controller for its various modes of operation is accomplished by writing various flag bits to the USER0 and USER1 registers. Additional information regarding the operation status of the MICRO PID Position Controller can be gleaned from the STATUS register. These three registers and their bit assignments are defined here.

Figure 25: USER0 Bits

Bit	Flag	Description	R/W
0	RetryFault	When set to "1" the controller will attempt to re-enable the H-bridge 250 PID periods after a fault condition occurs. This will continue indefinitely.	R/W
1	RunFromEEPROM	When set to "1" and configured for trapezoidal mode of operation the controller will run the trapezoidal movement profile stored in EEPROM on power up (the movement profile may be loaded into EEPROM with the **Store Profile** command).	R/W
2	TrapLogger	When set to "1" and configured for trapezoidal mode this flag causes the controller to send the Logging Mode Response (5.2.21) out of the TX pin every 20 PID periods. This flag is self-clearing and will clear when the trapezoidal movement profile is complete.	R/W
3	RestoreDefaults	When set to a "1" the factory default settings for all user accessible registers will be reset. This includes all trapezoidal segment data, PID settings, analog registers, velocity registers, and the user registers. This flag is self-clearing.	R/W
4	SoftwareReset	When set to a "1" a software reset occurs. This flag is self-clearing.	R/W
5	FourXMode	When set to a "1" the quadrature signal from the motor is decoded into 4 pulses for every set of input pulses on the channel A and channel B encoder lines (4X mode). When cleared ("0") each set of pulses from the quadrature signal is decoded into a single output pulse. See section 3.11 for more on 4X_1X mode of operation.	R/W
6	StepLogger	When set to "1" and **not configured** for trapezoidal mode this flag causes the controller to send the Logging Mode Response (5.2.21) out of the TX pin every 20 PID periods. The data will begin being sent when a **Write Desired Position** command is received. This flag is self-clearing and will clear when the Integral Clear Counter expires (therefore the motor must be "in position" for the flag to be cleared). If the motor does not reach the desired position sending a Write Position command will also clear the flag.	R/W
7	Unused		R/W

Figure 26: USER1 Bits

Bit	Flag	Description	R/W
0	AnalogMode	When set to a "1" Analog Control mode is enabled. Only 1 of bits 0 through 3 may be set at a time. The controller will clear all bits after the first bit it finds set.	R/W
1	TrapezoidalMode	When set to a "1" Trapezoidal Control mode is enabled. Only 1 of bits 0 through 3 may be set at a time. The controller will clear all bits after the first bit it finds set.	R/W
2	VelocityMode	When set to a "1" Velocity Control mode is enabled. Only 1 of bits 0 through 3 may be set at a time. The controller will clear all bits after the first bit it finds set.	R/W
3	DirectMode	When set to a "1" Direct Control mode is enabled. Only 1 of bits 0 through 3 may be set at a time. The controller will clear all bits after the first bit it finds set.	R/W
4	NegSWEnabled	When set to a "1" the negative limit switch function is enabled. If the input on the NEG_SWITCH (pin 17 of J5) is in the asserted state and NegSWEnabled is set to a "1" motor movement in the negative direction is prohibited.	R/W
5	PosSWEnabled	When set to a "1" the positive limit switch function is enabled. If the input on the POS_SWITCH (pin 16 of J5) is in the asserted state and PosSWEnabled is set to a "1" motor movement in the positive direction is prohibited.	R/W
6	NegSWState	When set to a "1" the asserted state of the negative limit switch is +5V. When set to a "0" the asserted state of the negative limit switch is 0V. This setting defaults to "0".	R/W
7	PosSWState	When set to a "1" the asserted state of the positive limit switch is +5V. When set to a "0" the asserted state of the positive limit switch is 0V. This setting defaults to "0".	R/W

Figure 27: STATUS Bits

Bit	Flag	Description	R/W
0	Fault	This bit is set to a "1" when an over-current or over-temperature fault occurs.	R
1	Position	This bit is set to a "1" when the actual position is equal to the desired position +/- the error-band setting.	R
2	NegativeSwitch	This bit is set to a "1" if the voltage on PIN17 of J5 matches the NegSWState setting.	R
3	PositiveSwitch	This bit is set to a "1" if the voltage on PIN16 of J5 matches the PosSWState setting.	R
4	Fault	This bit is set to a "1" when a fault occurs.	R
5	Unused		R
6	Unused		R
7	Unused		R

5.3.1 Write User Registers Example

To operate the MICRO PID Position Controller in Analog Mode, with automatic retry on a fault, 4X mode, and both limit switches enabled (asserted when the limit switches = 0V), the USER0 register would equal 0x21 (binary ' 00100001' ←LSB) and the USER1 register would equal 0x31 (binary '00110001'←LSB). The total message string sent to a MICRO PID Position Controller would look like this…

0xEB, 0x01, 0x21, 0x31, 0x3E

5.4 Changing Baud Rates

The MICRO PID Position Controller operates with a default serial data speed of 38.4KBPS. Since some devices may not be able to send and receive data at this rate accommodations have been made to allow the user to modify the baud rate.

Once modified the new baud rate takes effect immediately. No response accompanies these commands, and once received all future serial communication should occur at the new baud rate. The command must be received at the existing baud rate.

If your processor is unable to send data at the default setting of 38.4KBPS it is recommended that you download the position control software available at our web site and use it to modify the baud rate setting.

Restoring the default settings of the device will not modify the baud rate once it has been changed from the default setting of 38.4KBPS.

5.4.1 Set Baud Rate to 38.4KBPS

Description: This command changes the baud rate of serial communication.			
Component of Command	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Command	243	1	243 (0xF3)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	244 (0xF4)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
none			
Notes:			

5.4.2 Set Baud Rate to 19.2KBPS

Description: This command changes the baud rate of serial communication.			
Component of Command	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Command	244	1	244 (0xF4)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	245 (0xF5)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
none			
Notes:			

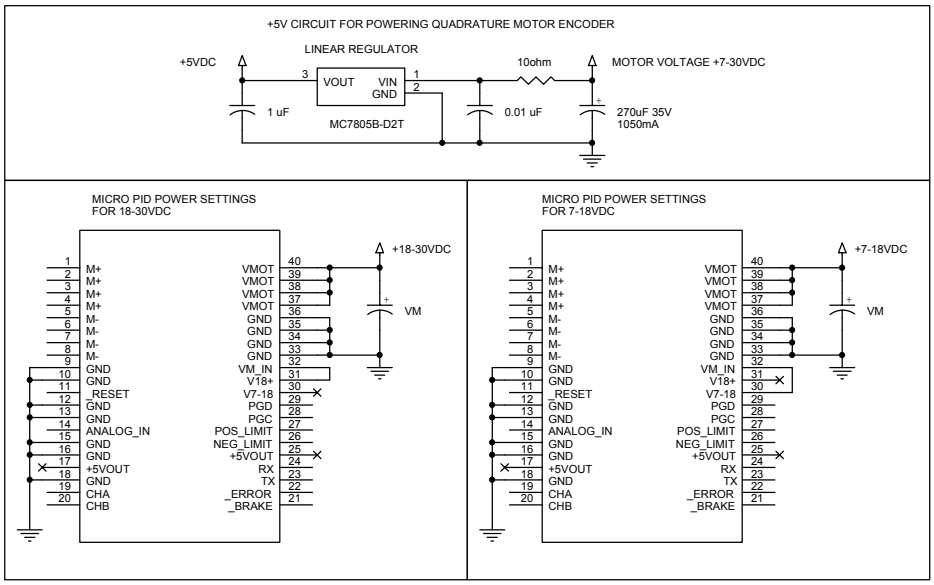
5.4.3 Set Baud Rate to 9.6KBPS

Description: This command changes the baud rate of serial communication.			
Component of Command	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
Command	245	1	245 (0xF5)
Address	0 to 255	1	1 (0x01)
Checksum	0 to 255	1	246 (0xF6)
Component of Response	**Range of Values**	**Number of Bytes**	**Decimal(Hex) Example**
none			
Notes:			

6. Quick Start

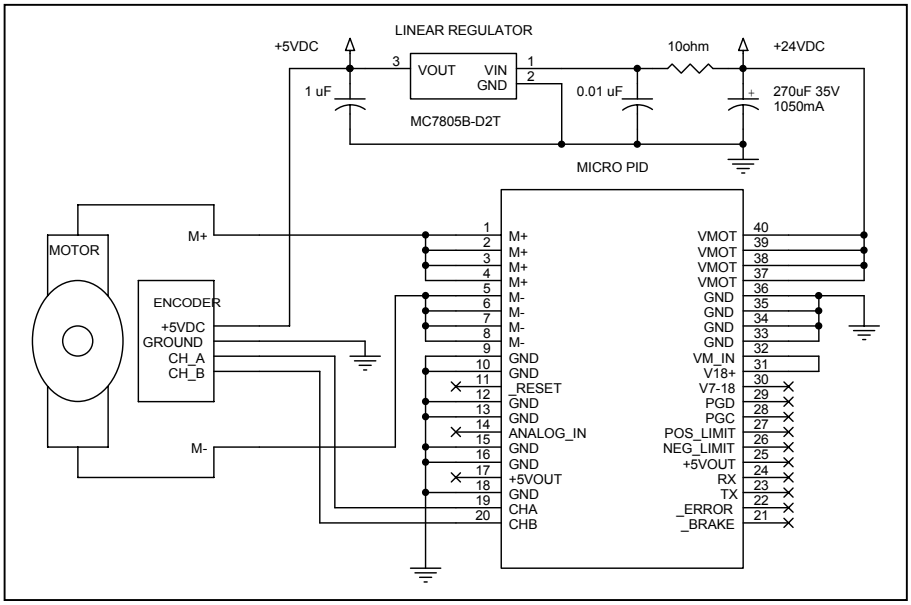
6.1.1 Select the Correct Power Setting

Connect your power supply and pins associated with powering the MICRO PID Position Controller as per the diagram below. The 5VOUT pins can provide a small amount of current to external loads. In general the MICRO PID 5V supply can not provide enough current to power a quadrature encoder. An external 5V supply is required to provide power to the quadrature encoder. The diagrams below can be used for reference. The 270uF capacitor below is the VM capacitor.



6.1.2 Connect your Motor and Encoder to the MICRO PID Position Controller

Connect your motor and encoder to the MICRO PID Position Controller as displayed below.

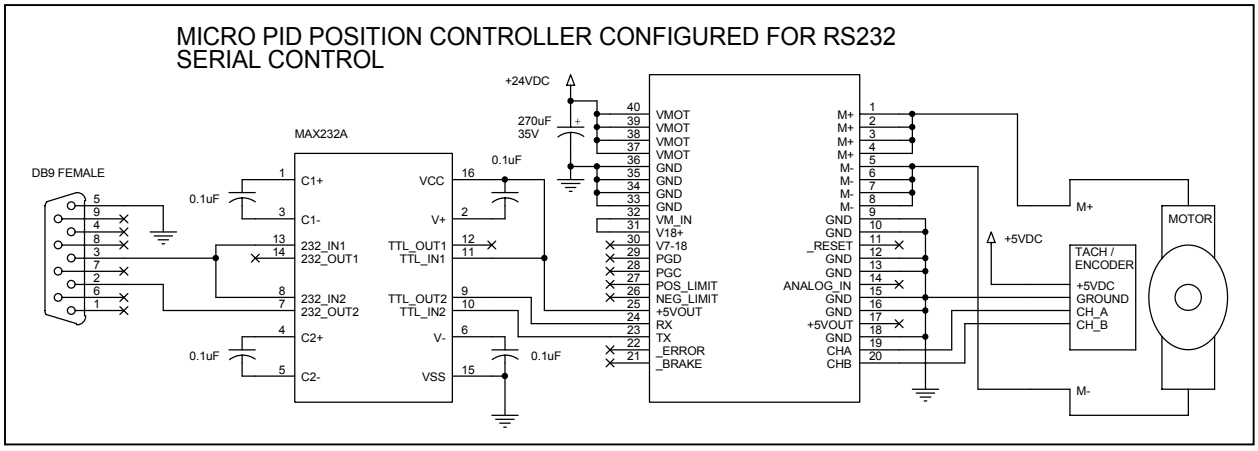


6.1.3 Apply Power to the System

When initially applying power to the position controller the “P” LED will blink once for analog mode, twice for trapezoidal mode, three times for velocity mode, and four times for direct position control. After the initial blinking the green “P” LED will stay lit if the motor and position controller are wired correctly. If the motor runs away either the motor leads are reversed or the quadrature encoder connections are not correct.

6.1.4 Quick Position Control

For fast access to the functionality available in the MICRO PID Position Controller you should download the position controller software available at www.solutions-cubed.com. You can also purchase a carrier board from Solutions Cubed that simplifies connectivity between a PC and the position controller. The MICRO PID Position Controller requires serial data at TTL levels (0-5VDC). In order to connect to a PC serial port conversion from RS232 to TTL levels and back must be addressed. A low cost option for connecting a PC to the position controller uses an off the shelf RS232 converter IC (such as the MAX232).



Note that the DB9 female connector shown above is a connector that would work if a standard DB9 cable were used between the circuit above and your PC’s serial port. The PC serial port is a DB9 male connector. To use this circuit with different connectors ensure that connections to the MAX232 pin labeled 232_IN carry data to the MICRO PID Position Controller. Data returned from the position controller are carried on the pin labeled 232_OUT. The MICRO PID Position Controller serial data pins describe their functionality in relation to the position controller (RX receives serial data, and TX sends serial data).

Disclaimer of Liability and Accuracy: Information provided by Solutions Cubed is believed to be accurate and reliable. However, Solutions Cubed assumes no responsibility for inaccuracies or omissions. Solutions Cubed assumes no responsibility for the use of this information and all use of such information shall be entirely at the user's own risk.

Life Support Policy: Solutions Cubed does not authorize any Solutions Cubed product for use in life support devices and/or systems without express written approval from Solutions Cubed.

Warrantee: Solutions Cubed warrants all Motor Control Modules against defects in materials and workmanship for a period of 90 days. If you discover a defect, we will, at our option, repair or replace your product or refund your purchase price. This warrantee does not cover products that have been physically abused or misused in any way.